Fuzzy Clustering with Knowledge Extraction and Granulation

Xianghui Hu, Yiming Tang, *Member, IEEE,* Witold Pedrycz, *Life Fellow, IEEE,*
Kai Di, Jiuchuan Jiang and Yichuan Jiang, *Senior Member, IEEE*

***Abstract*—Knowledge-based clustering algorithms can improve traditional clustering models by introducing domain knowledge to identify the underlying data structure. While there have been several approaches to clustering with the guidance of knowledge tidbits, most of them mainly focus on numeric knowledge without considering the uncertain nature of information. To capture the uncertainty of information, pure numeric knowledge tidbits are expanded to knowledge granules in this paper. Then two questions arise: how to obtain granular knowledge and how to use those knowledge granules in clustering. To the end, a novel knowledge extraction and granulation method and a granular knowledge-based fuzzy clustering model are proposed in this study. First, inspired by the concept of natural neighbors, an automatic knowledge extraction and granulation method (KEG) is developed. In KEG, high-density points are filtered from the dataset and then merged with their natural neighbors to form several dense areas, i.e., granular knowledge. Furthermore, the granular knowledge expressed by interval or triangular numbers is leveraged into the clustering algorithm, which is the framework of fuzzy clustering with granular knowledge. To concretize this model into clustering algorithms, the classical fuzzy C-Means (FCM) clustering algorithm has been selected to incorporate the granular knowledge produced by KEG. Then, the corresponding fuzzy C-Means clustering with interval knowledge granules (IKG-FCM) and triangular knowledge granules (TKG-FCM) are proposed. Experiments on synthetic and real-world datasets demonstrate that IKG-FCM and TKG-FCM always achieve better clustering performance with less time cost, especially on imbalanced data, compared with state-of-the-art algorithms.**

***Index Terms*—Fuzzy clustering, granular computing, knowledge extraction, unsupervised learning.**

I. Introduction

KNOWLEDGE-BASED clustering is an evolutionary branch of traditional clustering that stems from the adoption of domain knowledge, which can be applied to the areas of electronic commerce, bearing fault diagnosis, or image segmentation. The classical Fuzzy C-Means (FCM) algorithm [1], [2] and some improvements [3]–[5] perform clustering tasks based only on unlabeled data. In this sense, they are purely data-based algorithms. However, in knowledge-based clustering, some knowledge hints are additionally provided to support the clustering process. The effectiveness of the overall optimization process is enhanced by focusing on a certain search direction, thus reducing the overall search effort.

According to the source of knowledge, knowledge-based clustering can be divided into to two categories: clustering with existing knowledge and transfer clustering. In clustering with existing knowledge, users or experts have some typical values in the data in advance, such as maximum, minimum or even partial data structures. Then this domain knowledge can be added to the algorithm to optimize the search. Knowledge is transferred from a source domain to target domains in transfer clustering, which implies that some relevant features or knowledge about the clusters in the source domain are induced in advance by some learning procedures, and then the learned knowledge is available for the clustering task of the target domain.

A representative algorithm for clustering with existing knowledge is fuzzy clustering with viewpoints (V-FCM), in which knowledge is conceptualized as viewpoints [6]. These viewpoints are extreme data (e.g., minimal, maximum, medium) provided by the user and are regarded as an integral part of the data structure to achieve a personalized search. Tang et al. used the highest density point as one of the prototypes, and then, the possibilistic fuzzy clustering algorithm with a high-density viewpoint (DVPFCM) was established. The experiments in [7] showed that DVPFCM can locate the clustering centers more accurately under the guidance of the high-density point. In [6] and [7], knowledge hints (viewpoints) guide the algorithms to find the data structure and directly determine part of the clustering results. The flexibility of the algorithms is strongly limited in this way and neither algorithm considers the misleading roles of viewpoints.

The self-taught clustering (STC) proposed in [8] is the first transfer clustering algorithm based on mutual information. In [9], Jiang and Chung proposed a transfer learning version of spectral clustering. Deng et al. extended transfer learning to prototype-based clustering and developed two corresponding transfer prototype-based fuzzy clustering (TPFC) algorithms

This work was supported by the National Key Research and Development Program of China (No. 2019YFB1405000), the National Natural Science Foundation of China (Nos. 62076060, 61932007, 71971109, 62176083 and 61673156), the Key Research and Development Program of Jiangsu Province of China (No. BE2022157), and the Defense Industrial Technology Development Program (JCKY2021214B002). (*Corresponding author: Yichuan Jiang.*)

Xianghui Hu, Kai Di and Yichuan Jiang are with the School of Computer Science and Engineering, Southeast University, Nanjing, 211189, China (e-mail: xhhu306@163.com, dikai@seu.edu.cn, yjiang@seu.edu.cn).

Yiming Tang is with the Anhui Province Key Laboratory of Affective Computing and Advanced Intelligent Machine, School of Computer and Information, Hefei University of Technology, Hefei, Anhui, 230601, China, and also with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6R 2V4, Canada (e-mail: tym608@163.com).

Witold Pedrycz is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, T6R 2V4, Canada, with Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah, 21589, Saudi Arabia, and also with Systems Research Institute, Polish Academy of Sciences, Warsaw, 01-447, Poland (e-mail: wpedrycz@ualberta.ca).

Jiuchuan Jiang is with the School of Information Engineering, Nanjing University of Finance and Economics, Nanjing, 210023, China (e-mail: jcjiang@nufe.edu.cn).

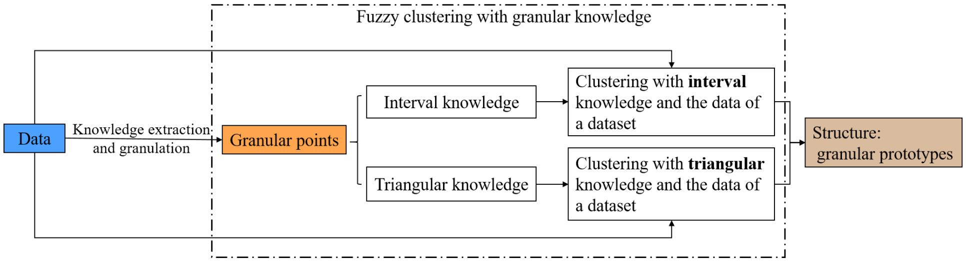

Fuzzy clustering with granular knowledge
Data
Knowledge extraction and granulation
Granular points
Interval knowledge
Triangular knowledge
Clustering with **interval** knowledge and the data of a dataset
Clustering with **triangular** knowledge and the data of a dataset
Structure: granular prototypes

Fig. 1. Framework of the fuzzy clustering with granular knowledge.

[10]. The key point of these algorithms is to fuse knowledge from the auxiliary dataset into the target dataset.

The existing knowledge-based clustering algorithms mainly adopt pure numeric knowledge data. However, in reality, due to the different cognitive degrees of people and the interference of the external environment, the description and understanding of knowledge are inconsistent and dynamic, which causes uncertainty in the obtained information [11]. On the other hand, knowledge itself has ambiguity, for example, there could be a dense area that characterizes a wide range and thus cannot be denoted by certain numeric datum [12]. Therefore, knowledge must be translated into granularity to make full use of the value of uncertain information. There are very few studies on clustering with granular knowledge, and only the fuzzy clustering with interval viewpoints (IV-FCM) proposed by [6] utilizes interval viewpoints to achieve a personalized search. However, the interval knowledge is still artificially specified and directly used as parts of the prototypes.

In summary, there are three problems in current knowledge-based clustering:

1) High cost of knowledge acquisition: The knowledge for clustering is either directly specified by users or extracted from auxiliary datasets [6], [8]–[10].
2) The limited form of knowledge: Most of the knowledge-based clustering algorithms (except for IV-FCM [6]) utilize pure numeric knowledge and do not capture the granular nature of information.
3) Overreaction of knowledge on clustering: In [6] and [7], knowledge tidbits directly replace parts of the prototypes, which severely limits the flexibility of the algorithms.

Those challenges provide new impetus to our work. To improve the efficiency of knowledge acquisition and to enhance the guidance of knowledge to the clustering process, we propose a novel automatic knowledge extraction and granulation (KEG) method as well as a fuzzy clustering framework based on granular knowledge.

First, the source of knowledge extraction is the dataset on which the clustering algorithm is carried out. In the proposed KEG method, we establish a new density calculation formula based on the concept of natural neighbors. Then, we utilize the outlier detection method three-sigma rule to automatically select high-density points. The natural neighbors around those high-density points are aggregated to form dense areas, which are denoted by interval or triangular numbers. These localized dense areas are knowledge granules. To explore the potential structure of the dataset, the data and granular knowledge are all available for clustering; see the framework of fuzzy clustering with granular knowledge in Fig. 1.

To the best of our knowledge, this article is the first work that investigates how to automatically obtain granular knowledge from a dataset and then how to play a leading role with these knowledge granules without overdoing it in clustering. One thing to note is that we are concerned with objective-function-based fuzzy clustering and FCM to make our investigation more focused and algorithmically tangible. This does not necessarily mean that the granular knowledge obtained by the proposed KEG method is restricted to merging with this particular category of fuzzy clustering. On the contrary, it could be easily extended to other objective-function-based clustering techniques such as possibilistic clustering or subspace clustering algorithms.

The remainder of this paper is arranged as follows. We start with a brief review of fuzzy clustering with viewpoints and the concept of granular information in Section II. Then, we move on to the details of the proposed knowledge extraction and granulation method KEG and two fuzzy clustering algorithms with knowledge granules KG-FCM (see Section III). Experimental results are shown in Section IV. Finally, we state the conclusions in Section V.

II. Related Work

Our study is closely related to the classic knowledge-based algorithm V-FCM and its granular version, fuzzy clustering with interval viewpoints (IV-FCM). Interval viewpoints are a model of information granules. We also briefly review the concept of information granules. Let us define $X = \{x_i\}_{i=1}^{n}$ as a sample set with n samples in d-dimensional space and suppose that the data X is divided into c clusters.

A. Fuzzy Clustering with Viewpoints

In [6], Pedrycz et al. creatively proposed a knowledge-driven clustering framework. They formalize domain knowledge as viewpoints. The viewpoints are provided by the user and are treated as externally introduced prototypes. Based on this approach, the original, data-based FCM algorithm is extended to V-FCM (FCM with viewpoints) and IV-FCM (FCM with interval viewpoints).

Viewpoints in [6] are defined by two matrix, denoted as B and F. The first one is a Boolean matrix and its elements are shown in the following form:

$$b_{jk} = \begin{cases} 1, & \text{if the } k\text{th feature of the } j\text{th prototype is} \\ & \text{determined by the viewpoint,} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The size of B is $c \times d$. The second matrix (F) is of the same dimensionality as B and includes the specific numeric values of the viewpoints, which is expressed as follows:

$$f_{jk} = \begin{cases} y, & b_{jk} = 1, \\ 0, & b_{jk} = 0, \end{cases} \tag{2}$$

where y is the value of the viewpoint. When viewpoints are interval data, $y = [y^L, y^R]$ where y^L and y^R are the left-hand side and right-hand side of y, respectively.

During the optimization of V-FCM and IV-FCM, the prototype v_{jk} is directly replaced by the value of a viewpoint when its corresponding b_{jk} is equal to 1. This is the most significant difference from FCM. The objective function is well-known one shown as follows.

$$v_{jk} = \begin{cases} \frac{\sum_{i=1}^{n} u_{ij}^m x_{ik}}{\sum_{i=1}^{n} u_{ij}^m}, & \text{if } b_{jk} = 0, \\ f_{jk}, & \text{if } b_{jk} = 1. \end{cases} \tag{3}$$

$$J = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^m \operatorname{dist}^2(v_j, x_i), \tag{4}$$

where $\operatorname{dist}^2(v_j, x_i)$ is the squared Euclidean distance between the point x_i and the jth clustering center v_j.

Both V-FCM and IV-FCM implement a personalized search for the structure because the viewpoints usually stand for what a user knows or is interested in. Inspired by this, we design a clustering algorithm guided by high-density granules to discover data structural features more accurately.

B. Information Granules

The basis of our work is the modelling of dense areas in a dataset as information granules. Generally, information granules are collections of entities that are arranged together due to their closeness (similarity, coherency, functionality, and so on.). There are three main categories of information granules: HFD [13], interval data and LR-type fuzzy numbers. An interval number is a range that consists of upper and lower bounds, and all numbers in this range have the same properties. An interval datum x is usually formalized as $[x^L, x^R]$. LR-type fuzzy numbers are a convenient and practical fuzzy data type [14]. There are many different classes of fuzzy numbers in applied problems, such as triangular [15]–[20], trapezoidal [15], [21]–[25], Gaussian [15] fuzzy numbers, etc.

Although there have been different granule types, most studies are based on the assumption that granular data are given a priori, except for Gacek and Pedrycz [26] and Shen et al. [27], who gave a mechanism to create granules. They form information granules with the use of the principle of justifiable granularity (PJG). The clustering algorithm proposed in [27] runs on granules transformed from numeric data by PJG. Therefore, their works belong to the clustering of granular data. In our study, we aim at knowledge extraction from a numeric dataset and knowledge granulation and the way the knowledge granules work in clustering numeric data.

III. FUZZY CLUSTERING WITH KNOWLEDGE GRANULES

In this section, a knowledge extraction and granulation algorithm (KEG) is established, and a fuzzy clustering algorithm with knowledge granules (KG-FCM) is presented.

A. Knowledge Extraction and Granulation

The purpose of the KEG algorithm is to automatically determine high-density areas of a dataset. Those areas are modeled as granular knowledge. Then, the KG-FCM algorithm will find a potential data structure with the guidance of granular knowledge. The KEG method mainly includes a data density measurement and granular knowledge formation. These two parts are both based on natural neighbors.

1) Natural neighbors:

The approach of natural neighbors is an interesting concept about neighbors. This concept simulates the friendship of human society in which the number real friends that a person has should be the number of people who take him or her as friends and he or she takes them as friends at the same time [28]. In a dataset, a natural neighbor x_j of a point x_i is defined as follows:

$$x_j \in NN_\theta(x_i) \Leftrightarrow ((x_i \in KNN_\theta(x_j)) \wedge (x_j \in KNN_\theta(x_i))) \tag{5}$$

where $NN_\theta(x_i)$ is a natural neighbor set of x_i for $i = 1, \dots, n$, and $KNN_\theta(x_i)$ is a set of the first to θ-th nearest neighbors of x_i. The $\wedge$ means the logical "and". If x_i is one of $KNN_\theta(x_j)$ and x_j is one element of $KNN_\theta(x_i)$, then x_j is a natural neighbor of x_i.

$$KNN_\theta(x_i) = \bigcup_{k=1}^{\theta} \{knn(x_i, k)\} \tag{6}$$

where $knn(x_i, k)$ is the k-th nearest neighbor of x_i. The distance between points is the Euclidean one.

The overall natural neighbor (NaN) method is summarized in Algorithm 1. The main procedure of the NaN method is to continuously expand the neighbor search range, see steps 3 to step 20. The times at which each data point is considered to be the other data point's neighbor are recorded at every expansion, see step 6. This computing in the loop stop when all objects have natural neighbors or the number of objects without natural neighbors does not change, see steps 15 to 18. In Algorithm 1, nb records the number of natural neighbors for each sample, which is an $n * 1$ array. The NaN method can effectively determine the neighborhood in a dataset without any pregiven parameters. When the neighbor search range θ is predefined, the concept of a natural neighbor is the mutual near neighbor mentioned in [29] and [30].

2) Knowledge extraction:

The idea of the KEG algorithm is based on the assumption that the distance between points in dense regions is smaller than that in sparse regions, and points lying in dense regions possess more neighbors than those in sparse regions. Thus, we define the density of a data point x_i as

$$\rho_i = \kappa \operatorname{min\_max}(\rho_{i1}) + (1 - \kappa) \operatorname{min\_max}(\rho_{i2}), \tag{7}$$

where

$$\rho_{i1} = \begin{cases} 0, & nb(x_i) = 0, \\ \frac{nb(x_i)}{\underset{x_j \in NN_\theta(x_i)}{mean} (\operatorname{dist}(x_i, x_j))}, & nb(x_i) \neq 0, \end{cases} \tag{8}$$

$$\rho_{i2} = \sum_{i \neq j} \exp\left[-\left(\frac{\operatorname{dist}(x_i, x_j)}{r}\right)^2\right]. \tag{9}$$

Algorithm 1 Natural Neighbor algorithm (NaN)

```
Input: A set of $n$ data points $X = \{x_i\}_{i=1}^{n}$
Output: $nb, NN_\theta$
procedure NAN(Data $X$)
    Initialization: $\theta = 1$, $nb(i) = 0$, $NN_0 = \varnothing$, $KNN_0 = \varnothing$;
    while $true$ do
        for all $x_i \in X$ do
            Find the $\theta$-th nearest neighbor $y$ of $x_i$;
            $KNN_\theta(x_i) = KNN_{(\theta-1)}(x_i) \cup \{y\}$;
            if $x_i \in KNN_\theta(y)$ & $x_i \notin NN_\theta(y)$ then
                $nb(y) = nb(y) + 1$;
                $NN_\theta(y) = NN_{(\theta-1)}(y) \cup \{x_i\}$;
                $nb(x_i) = nb(x_i) + 1$;
                $NN_\theta(x_i) = NN_{(\theta-1)}(x_i) \cup \{y\}$;
            end if
        end for
        ▷ Calculate the number of points with no neighbors
        $Num_\theta = count(nb(x_i) == 0)$;
        if $Num_{(\theta-1)} == Num_\theta$ or $Num_\theta == 0$ then
            break;
        end if
        $\theta = \theta + 1$;
    end while
    return $nb$, $NN_\theta$;
end procedure
```

Operator $\operatorname{dist}(x_i, x_j)$ is the Euclidean distance between samples x_i and x_j. The function $\min\_\max$ of (7) is the min-max normalization to scale the ρ_{i1} and ρ_{i2} range in $[0, 1]$ to eliminate scale differences between them. From (7), we know that the local density ρ_i of point x_i is a sum of its natural-neighbor density ρ_{i1} and kernel distance-based density ρ_{i2}. Parameters κ and $1 - \kappa$ could be considered as the relative importance factors for two density indices. Here, the constant κ is in $[0, 1]$. If $0 \leq \kappa < 0.5$, there is a higher importance to the kernel distance-based density ρ_{i2}, and when $0.5 < \kappa \leq 1$, the natural-neighbor density ρ_{i1} has a higher importance. We assign equal importance to ρ_{i1} and ρ_{i2} in measuring the local density of data x_i, so $\kappa = 0.5$.

To make the extracted high-density points more representative, we compute the relative distance δ_i between x_i and points with larger local density.

$$\delta_i = \begin{cases} \min\limits_{j:\rho_j > \rho_i} \left(\operatorname{dist}(x_i, x_j)\right), & \exists j \text{ s. t. } \rho_j > \rho_i, \\ \max_j \left(\operatorname{dist}(x_i, x_j)\right), & \text{otherwise.} \end{cases} \tag{10}$$

It is indicated in (8) that the natural-neighbor density ρ_{i1} of x_i is inversely proportional to the mean distance between itself and its natural neighbors and proportional to the number of natural neighbors. Because two different points can have the same number of neighbors, we also consider the mean distance between the point and its natural neighbors, which represents the relative dispersion around the data x_i. The smaller the sum of the neighbors' distance and the larger the number of natural neighbors, the denser the region the point lies in (respectively, the larger and smaller, the sparser).

Parameter ρ_{i2} is also the kernel distance-based local density as defined in [31]. However, to obtain an appropriate density assessment range and consider the distribution of a dataset, we calculate the radius r as follows:

$$r = \max(\operatorname{dist}(x_i, x_j))/(2c), \ i, j \in \{1, 2, \cdots, n\}. \tag{11}$$

We suppose that all points are contained in a hypersphere. The diameter of this hypersphere is the maximum distance between points (i.e., $\max(\operatorname{dist}(x_i, x_j))$) divided by the double cluster number c. In other words, this hypersphere is composed of c small spheres. Therefore, the distance between points in each cluster does not exceed r. This concept is useful in delineating the cutoff distance proved by [7].

Compared to some density-based methods ([31], [32]), the proposed KEG method is parameter-free without preset radius r and neighbor number k. KEG computes local density from two aspects: natural neighbors and specific regional neighbors.

In clustering by fast search and finding density peaks (DPC) and some improvement algorithms, the investigators obtain cluster centers (i.e., more high-density points) by hand-extraction from its $\rho - \delta$ decision graph [31], [33]–[35]. This method is inconvenient and vulnerable to personal judgments. We take the Aggregation dataset in [36] as an example, and its reference clustering results are shown in Fig. 2(a). Fig. 2(c) is the $\rho - \delta$ decision graph of DPC for the Aggregation dataset. We can see from Fig. 2(c) that the minimum interval between the points covered by the red grid and the blue dots, and the minimum gap between them and the other black dots are both relatively small. So whether the points covered by the red grid should be selected as high-density points is undecidable. In addition, from Fig. 2(b), we can observe that the δs values of potential knowledge points shown by the colored dots deviate significantly from the others (black dots).

In this paper, we first sort δ in ascending order to obtain δ' and its corresponding data order $X' = \{x'_i\}_{i=1}^{n}$, see Algorithm 2 step 15. Then we measure the relative separation in δ's by the difference ϕ between adjacent elements of δ', see steps 16-18. The characteristic of high-density points is that their δ values are larger, which causes the ϕ between them and non-density points to be larger as well. These larger ϕs are outliers relative to others. We use the commonly used outlier detection method three-sigma rule to detect ϕ outliers automatically [37], see steps 19 to 25 in Algorithm 2. Based on the three-sigma rule, the definition of the ϕ outlier is presented as follows.

Definition 1. *Parameter ϕ_i belongs to an outlier if its value is more than three standard deviations from the mean, i.e.*

$$\begin{aligned} \left|\phi_i - \overline{\Phi}\right| &\geq 3\sigma_{diff}, \\ \overline{\Phi} &= \frac{1}{n-1}\sum_{i=1}^{n-1}\phi_i, \\ \sigma_{diff} &= \sqrt{\frac{1}{n-2}\sum_{i=1}^{n-1}\left|\phi_i - \overline{\Phi}\right|^2} \end{aligned} \tag{12}$$

where $\overline{\Phi}$ and σ_{diff} are the mean and the standard deviation of ϕs, respectively.

The larger values ϕs can be used as the dividing lines between high-density and non-high-density points. For example, if ϕ_i is significantly larger relative to other ϕs, then $\{x'_{i+1}, x'_{i+2}, \cdots, x'_n\}$ are high-density points candidates

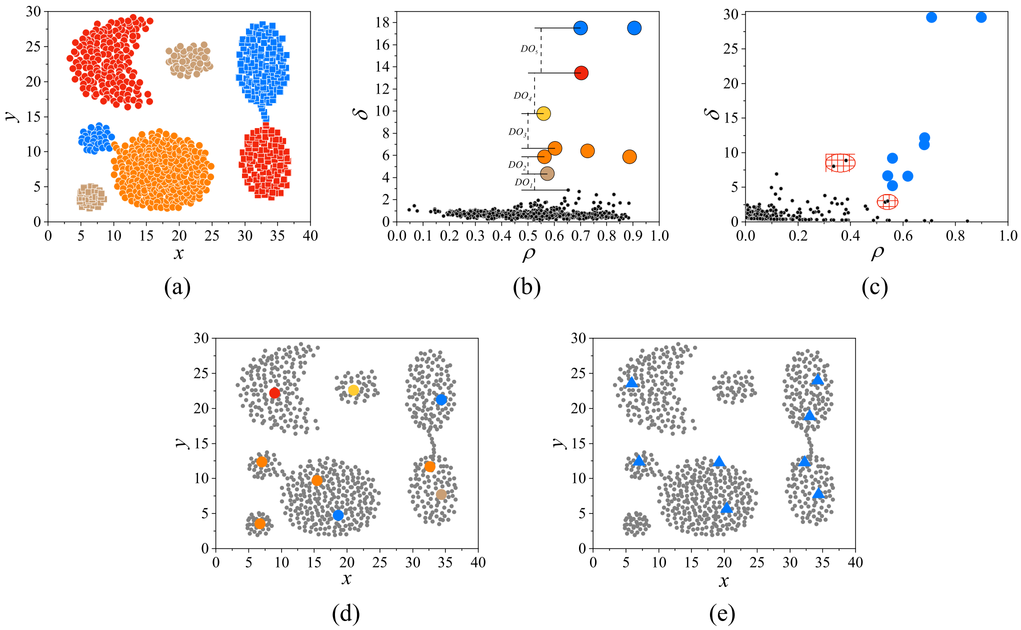


Fig. 2. Dataset Aggregation. (a) Data distribution. Different colors and shapes represent different clusters. (b) Decision graph of KEG. Different colors indicate that the points are located between different division lines DOs. (c) Decision graph of DPC. (d) High-density points selected by KEG. (e) High-density points selected by DPC.

(i.e. Knowledge points set G in step 21 of Algorithm 2), while $\{x_1', \cdots, x_i'\}$ are non-high-density points candidates. Although the three-sigma rule requires that the data obey normal distribution, according to Chebyshev's inequality, for non-normally distributed data, at least 88 percent of data are in the range of three times sigma [38], [39].

If there are noises in the dataset, such as points 21, 22 and 23 shown in Fig. 3(a), they also have a relatively high δ. But their ρs are low because they are isolated. To avoid the δ of noises data interfering with the detection of ϕ outliers, we set the δ value of the low-density points (see Definition 2) to the average of δ, i.e., step 10 to 14 in Algorithm 2.

Definition 2. *if $\rho_i \leq \overline{\rho} - \sigma_{dens}$, x_i is a point with low local density.*

$$\overline{\rho} = \frac{1}{n}\sum_{i=1}^{n}\rho_i, \qquad (13)$$
$$\sigma_{dens} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left|\rho_i - \overline{\rho}\right|^2}.$$

For the Aggregation dataset, five ϕ outliers can be found based on (12), i.e. DO_1 to DO_5 in Fig. 2(b). This means that there are five dividing lines between high-density and non-high-density points, and their corresponding ϕs are the line widths. The data points whose δs are over the line are high-density points. All high-density points can be acquired based on DO_1. Finally, the KEG method automatically obtains nine high-density points, as shown in Fig. 2(d). The DPC method manually gains eight high-density points. KEG makes each cluster have at least one high-density point, but there is a cluster without high-density points obtained by DPC. In contrast, there are two clusters that have two high-density points each in Fig. 2(e). Overall, the proposed KEG method can conveniently locate high-density knowledge more accurately.

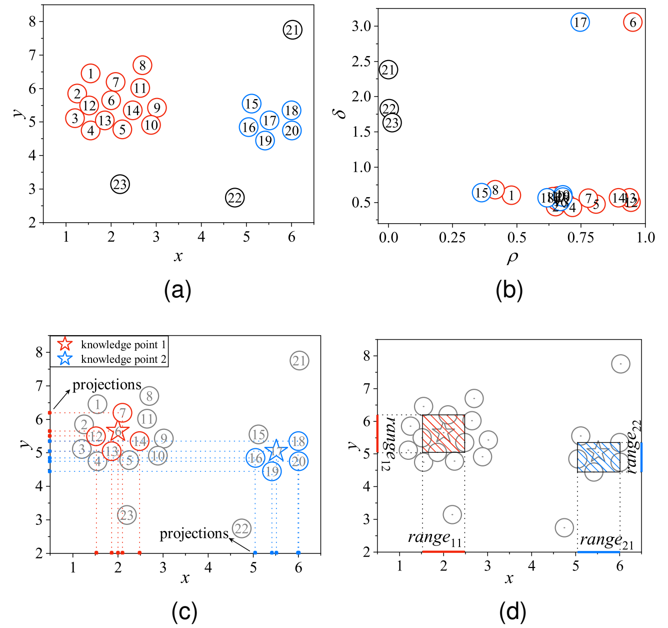


Fig. 3. The dataset used to illustrate Algorithm 2. (a) Data distribution. Data points are ranked randomly. Different colors represent different clusters. (b) Decision graph of KEG. (c) Building knowledge granules. (d) Range limits (rectangle) for knowledge granules.

3) Knowledge granulation:

After extracting the high-density knowledge points, we granulate those points to represent dense areas where they are located. We adopt the two types of information granules, interval and triangular data. Interval and triangular knowledge granules are both modelled as hyperboxes, inside which the high-density point and its natural neighbors are confined. The

Algorithm 2 Knowledge extraction and granulation algorithm (KEG)

```
Input: A set of $n$ data points $X=\{x_i\}_{i=1}^n$, cluster number $c$,
    granular type $GT$
Output: Knowledge points $G$, granular knowledge $IG$ or $TG$
 1: procedure KEG(Data $X$, Granular type $GT$)
 2:     $(nb, NN_\theta)=\mathrm{NaN}(X)$;
 3: ▷ Knowledge extraction
 4:     for $i=1:n$ do
 5:         Calculate $\rho_{i1}$ using (8);
 6:         Calculate $\rho_{i2}$ using (9);
 7:         Calculate $\rho_i$ using (7);
 8:         Calculate $\delta_i$ using (10);
 9:     end for
10:     for $i=1:n$ do
11:         if $\rho_i \le \overline{\rho} - \sigma_{density}$ then
12:             $\delta_i = \overline{\delta}$;
13:         end if
14:     end for
15:     Sort $\delta=\{\delta_i\}_{i=1}^n$ in ascending order to get $\delta'=\{\delta'_i\}_{i=1}^n$
    and the dataset $X'=\{x'_i\}_{i=1}^n$ corresponding to the set $\delta'$;
16:     for $i=1:n-1$ do
17:         Calculate $\phi_i=\delta'_{i+1}-\delta'_i$;
18:     end for
19:     for $i=1:n-1$ do
20:         if $\left|\phi_i-\overline{\Phi}\right| \ge 3\sigma_{\text{diff}}$ then
21:             Knowledge points set $G=\{g_k|g_k=x'_{i+k}\}_{k=1}^{n-i}$;
22:             The number of knowledge points $K=n-i$;
23:             break;
24:         end if
25:     end for
26: ▷ Extracted knowledge granulation
27:     if $GT=1$ then
28:         Granulate the knowledge set $G$ using (17) for
    interval knowledge $IG$;
29:     else
30:         Granulate the knowledge set $G$ using (19) for
    triangular knowledge $TG$;
31:     end if
32:     return $G=\{g_k\}_{k=1}^K$, granular knowledge $IG$ or $TG$;
33: end procedure
```

data covered by the kth knowledge granule is denoted as

$$co_k=\{g_k, NN_\theta(g_k)\}, k=1,\ldots,K. \tag{14}$$

The side length of a hyperbox equals the range of each feature. Here, the range of the jth feature for the kth knowledge granule is defined as

$$range_{kj}=\max_{s=1,\ldots,n'}(co_{ksj})-\min_{s=1,\ldots,n'}(co_{ksj}) \tag{15}$$

where co_{ksj} is the projection of the sth data point covered by the kth knowledge granule on the jth feature and n' is the number of data in the hyperbox.

Once the data contained in the hyperbox have been determined, we can get an interval knowledge granule set $IG=\{ig_1,\ldots,ig_K\}$ where ig_k is represented by its bounds as

$$ig_k=[ig_k^L, ig_k^R] \text{ for } k=1,\ldots,K \tag{16}$$

where ig_k^L and ig_k^R are the left and right side of the kth interval knowledge granule, respectively. They are vectors in $\mathbb{R}^d$ expressed in the form

$$\begin{aligned} ig_k^L&=[\min_{s=1,\ldots,n'}(co_{ks1}),\ldots,\min_{s=1,\ldots,n'}(co_{ksd})]^{\mathrm{T}},\\ ig_k^R&=[\max_{s=1,\ldots,n'}(co_{ks1}),\ldots,\max_{s=1,\ldots,n'}(co_{ksd})]^{\mathrm{T}}. \end{aligned} \tag{17}$$

Similarly, a triangular knowledge granule set $TG=\{tg_1,\ldots,tg_K\}$ can be obtained where tg_k is characterized by a triple.

$$tg_k=(tg_k^L, tg_k^M, tg_k^R) \text{ for } k=1,\ldots,K \tag{18}$$

where tg_k^L, tg_k^M and tg_k^R are the left, middle and right side of the kth triangular knowledge granule, respectively. They are also vectors in $\mathbb{R}^d$ represented as

$$\begin{aligned} tg_k^L&=[\min_{s=1,\ldots,n'}(co_{ks1}),\ldots,\min_{s=1,\ldots,n'}(co_{ksd})]^{\mathrm{T}},\\ tg_k^M&=[g_{k1},g_{k2},\ldots,g_{kd}]^{\mathrm{T}},\\ tg_k^R&=[\max_{s=1,\ldots,n'}(co_{ks1}),\ldots,\max_{s=1,\ldots,n'}(co_{ksd})]^{\mathrm{T}}. \end{aligned} \tag{19}$$

To better understand the overall process of knowledge granulation, we illustrate it in Fig. 3(c) and (d). In Fig. 3(c), two knowledge points 6 and 17 (denoted by red and blue pentagrams, respectively) are selected by steps 19 to 25 in Algorithm 2. Then the knowledge points and their natural neighbors (denoted by red and blue hollow circles) are used to construct interval and triangular knowledge granules by (17) and (19) resulting from the projections (denoted by red and blue solid dots) on each feature, respectively. Finally, two rectangles are formed, inside which we see $n'=5$ points are confined from Fig. 3(d).

Here, the knowledge extraction and granulation framework is established, as shown in Algorithm 2. It can automatically obtain high-density points and granulate its corresponding dense areas to enrich the power of knowledge. However, as shown in Fig. 2(d), the KEG method obtains 9 high-density points, which is above the true number of clusters, 7. This number means that the KEG cannot accurately locate all dense areas, which will also be shown later in experimental section IV. Therefore, we propose a clustering algorithm with granular knowledge to accomplish the clustering task. The details are shown in the next section.

B. Objective Function

Based on the proposed KEG knowledge extraction and granulation method, we can obtain K dense areas represented by IG or TG. To exploit granular knowledge to help the clustering task in a dataset, we develop a new fuzzy clustering algorithm, fuzzy C-Means clustering with knowledge granules (KG-FCM). The objective function of KG-FCM is defined by

$$\begin{aligned} J_{\text{KG-FCM}}=&\sum_{i=1}^{n}\sum_{j=1}^{c}u_{ij}^m\eta_i d_{\text{KG}}^2(v_j,x_i)+\zeta\left(\sum_{j=1}^{c}\sum_{k=1}^{K}\omega_{jk}d_{\text{KG}}^2(v_j,gg_k)\right.\\ &\left.+\sum_{j=1}^{c}\sigma_j\sum_{k=1}^{K}(\omega_{jk}\ln\omega_{jk}-\omega_{jk})\right) \end{aligned} \tag{20}$$

subject to

$$\sum_{j=1}^{c} u_{ij} = 1, 0 \leqslant u_{ij} \leqslant 1, 0 \leqslant \eta_i \leqslant 1, 0 \leqslant \omega_{jk} \leqslant 1. \tag{21}$$

In (20),

$$\sigma_j = \operatorname{dist}\left(v_j, \bar{G}\right)^2, \ \bar{G} = \frac{\sum_{k=1}^{K} g_k}{K}. \tag{22}$$

$$\eta_i = 1 - \frac{\min\limits_{k=1,\ldots,K}\left(\operatorname{dist}\left(x_i, g_k\right)\right)}{\max\limits_{j=1,..,n}\left(\min\limits_{k=1,\ldots,K}\left(\operatorname{dist}\left(x_j, g_k\right)\right)\right)}. \tag{23}$$

Because the KEG method can provide two types of knowledge, we introduce two variants of the KG-FCM algorithms: fuzzy C-Means clustering with interval knowledge granules (IKG-FCM) and triangular knowledge granules (TKG-FCM). Their objective functions are similar to (20), but the distance measurements are changed. d_{KG} is represented by d_{IG} and gg_k is ig_k in IKG-FCM, while d_{KG} is d_{TG} and gg_k is tg_k in TKG-FCM. The specific computations are shown in (24) and (25).

$$d_{\mathrm{IG}}^2(v_j, ig_k) = \sum_{t=1}^{d} \frac{(v_{jt}^L - ig_{kt}^L)^2 + (v_{jt}^R - ig_{kt}^R)^2}{2}, \tag{24}$$

$$\begin{aligned} & d_{\mathrm{TG}}^2\left(v_j, tg_k\right) \\ = & \sum_{t=1}^{d}\left(\frac{\alpha}{6}\left(v_{jt}^L - tg_{kt}^L\right)^2 + \left(1 - \frac{5}{3}\alpha\right)\left(v_{jt}^M - tg_{kt}^M\right)^2\right. \\ & + \frac{\alpha}{6}\left(v_{jt}^R - tg_{kt}^R\right)^2 + \frac{\alpha}{6}\left(v_{jt}^L - tg_{kt}^L + v_{jt}^M - tg_{kt}^M\right)^2 \\ & \left. + \frac{\alpha}{6}\left(v_{jt}^R - tg_{kt}^R + v_{jt}^M - tg_{kt}^M\right)^2\right) \end{aligned} \tag{25}$$

with $0 < \alpha < 0.5$.

Some explanations of (20) are given as follows:

1) The first term is an improvement of the classic FCM in which η_i is a weight associated with data x_i. The data weight η_i is expressed in (23). The quantity $\max\limits_{j=1,..,n}(\min\limits_{k=1,\ldots,K}(\operatorname{dist}(x_j, g_k)))$ serves as a normalization coefficient that keeps the values of the weight confined to the unit interval. The relationship in (23) shows that the closer the data are to the knowledge points, the higher their contribution to the clustering process.
2) The second term is used to bring on granular knowledge from the KEG method to guide the clustering task. The parameter ζ balances the influence of granular knowledge and is a nonnegative value. Parameter analysis will be shown in Section IV. In this term, ω_{jk} weights the influence of the granular knowledge gg_k on the clustering center v_j. Note that the right-hand side of the second term prevents the trivial zero solution for ω_{jk}s.
3) In IKG-FCM, centers and knowledge points are interval values, i.e., $v_j = [v_j^L, v_j^R]$ and $ig_k = [ig_k^L, ig_k^R]$. $d_{\mathrm{IG}}^2(v_j, ig_k)$ takes the Euclidean distance, shown in (24). In addition, for the triangular data, we use the integral metric proposed in [40]. The weights of the left-hand side, middle side and right-hand side of triangular data are measured by α_1,α_2 and α_3 in this metric. Generally, we constrain $\alpha_1 + \alpha_2 + \alpha_3 = 1$, $\alpha_1 > 0$, $\alpha_2 > 0$ and $\alpha_3 > 0$. In TKG-FCM, the left-hand side and right-hand side of triangular knowledge $tg_k = (tg_k^L, tg_k^M, tg_k^R)$ or center $v_j = (v_j^L, v_j^M, v_j^R)$ possess the same importance, and the middle side is more representative than ends. Thus, $\alpha_2 > \alpha_1 = \alpha_3 = \alpha$ $(0 < \alpha < 0.5)$ and $\alpha_2 = 1 - 2\alpha$. We set α to 0.25 in the experiment. The form of $d_{\mathrm{TG}}^2(v_j, tg_k)$ is denoted by (25). In our algorithms, point x_i is treated as a special case of an interval or triangular number. Therefore, the values of its left, middle and right sides are equal to the numeric values.

According to the optimization strategy that invokes the Lagrange multipliers, we can obtain the following update rules for KG-FCM:

$$u_{ij} = \frac{d_{\mathrm{KG}}\left(v_j, x_i\right)^{\frac{-2}{m-1}}}{\sum\limits_{j'=1}^{c} d_{\mathrm{KG}}\left(v_{j'}, x_i\right)^{\frac{-2}{m-1}}}, \tag{26}$$

$$\omega_{jk} = \exp\left(-\frac{d_{\mathrm{KG}}\left(v_j, gg_k\right)^2}{\sigma_j}\right). \tag{27}$$

The centers v_js in IKG-FCM are updated by:

$$v_j^L = \frac{\sum\limits_{i=1}^{n} u_{ij}^m \eta_i x_i + \zeta \sum\limits_{k=1}^{K} \omega_{jk} + ig_k^L}{\sum\limits_{i=1}^{n} u_{ij}^m \eta_i + \zeta \sum\limits_{k=1}^{K} \omega_{jk}}, \tag{28}$$

$$v_j^R = \frac{\sum\limits_{i=1}^{n} u_{ij}^m \eta_i x_i + \zeta \sum\limits_{k=1}^{K} \omega_{jk} ig_k^R}{\sum\limits_{i=1}^{n} u_{ij}^m \eta_i + \zeta \sum\limits_{k=1}^{K} \omega_{jk}}. \tag{29}$$

Meanwhile, in TKG-FCM, updating equations for the cluster centers v_j are as follows:

$$v_j^L = \frac{\sum\limits_{i=1}^{n} u_{ij}^m \eta_i \left(3x_i - v_j^M\right) + \zeta \sum\limits_{k=1}^{K} \omega_{jk}\left(2tg_k^L + tg_k^M - v_j^M\right)}{\sum\limits_{i=1}^{n} 2u_{ij}^m \eta_i + 2\zeta \sum\limits_{k=1}^{K} \omega_{jk}}, \tag{30}$$

$$v_j^M = \frac{\sum\limits_{i=1}^{n} u_{ij}^m \eta_i \left(\left(1 - \frac{2}{3}\alpha\right) x_i - \frac{\alpha}{3}\left(v_j^L + v_j^R\right)\right) + \zeta \sum\limits_{k=1}^{K} \omega_{jk} tmp}{\left(1 - \frac{4}{3}\alpha\right)\left(\sum\limits_{i=1}^{n} u_{ij}^m \eta_i + \zeta \sum\limits_{k=1}^{K} \omega_{jk}\right)} \tag{31}$$

where the temporary variable tmp is

$$tmp = \left(1 - \frac{4}{3}\alpha\right) tg_k^M + \frac{\alpha}{3}(tg_k^L - v_j^L) + \frac{\alpha}{3}(tg_k^R - v_j^R). \tag{32}$$

$$v_j^R = \frac{\sum\limits_{i=1}^{n} u_{ij}^m \eta_i (3x_i - v_j^M) + \zeta \sum\limits_{k=1}^{K} \omega_{jk}(2tg_k^R + tg_k^M - v_j^M)}{\sum\limits_{i=1}^{n} 2u_{ij}^m \eta_i + 2\zeta \sum\limits_{k=1}^{K} \omega_{jk}}. \tag{33}$$

We use the Euclidean distance and the integral metric to measure the distance between two interval data and two triangular data, respectively, and as a result, the centers updating form for IKG-FCM and TKG-FCM have different forms.

C. The Clustering Algorithm and Complexity Analyses

1) Clustering algorithm:

The model KG-FCM is summarized as Algorithm 3.

2) Convergence of the KG-FCM algorithm:

Based on Zangwill's convergence theorem, Lagrange's theorem [55] and the bordered Hessian matrix [56], the proofs of the KG-FCM convergence is presented in Appendix-I.

3) Analyses of the computational complexity:

Given the input dataset with n samples, c clusters and T running iterations, there are two main parts for the computational complexity of KG-FCM. The first part is knowledge extraction and its granulation, i.e., the KEG method. Its main cost of computation is computing the distance between n samples, and thus, the time complexity of this part is $\mathcal{O}(n^2)$. The second part is for the alternation training. The computational complexity of this part is $\mathcal{O}(ncT)$. Thus, the overall cost for KG-FCM is $\mathcal{O}(n^2+ncT)$. Generally, the iteration number T and the cluster number c are far smaller than samples number n, and the computational complexity of KG-FCM can be rewritten as $\mathcal{O}(n^2)$. The computational cost focuses on the data distance $\operatorname{dist}(x_i, x_j)(i, j = 1, \cdots, n)$. The data distance does not change dynamically, and thus, we can precompute it and store it in advance. When running the KEG method, this information is loaded from the hard disk. This method will greatly reduce the running time of the KG-FCM algorithm.

Algorithm 3 Fuzzy clustering algorithm based on granular knowledge (KG-FCM)

```
Input: A set of $n$ data points $X=\{x_i\}_{i=1}^{n}$, the number
    of clusters $c$, the fuzzy index $m$, the parameter $\zeta$, the
    termination criterion $\epsilon$ and $T$, and granular type $GT$
Output: The cluster center matrix $V=\{v_j\}_{j=1}^{c}$, the partition
    matrix $U=\{u_{ij}\}_{i,j=1}^{n,c}$
procedure KG-FCM(Data $X$, Number $c$, Type $GT$)
▷ Get granular knowledge
    if $GT = 1$ then
        $[G, IG] = \mathrm{KEG}(X, GT)$;
    else
        $[G, TG] = \mathrm{KEG}(X, GT)$;
    end if
    Randomly select $c$ data from $X$ as initial cluster
  centers $V^{(0)}$;
    Calculate data weight $\eta_i$ using (23) for $i = 1, \ldots, n$;
    Set the number of iterations $t=0$;
▷ Iterative training
    repeat
        Calculate $U^{(t)}$ by (26);
        Calculate $\omega_{jk}$ using (27);
        if $GT = 1$ then
            Calculate $V^{(t+1)}$ using (28) and (29);
        else
            Calculate $V^{(t+1)}$ using (30) to (33);
        end if
        Update $t=t+1$;
    until $|J^{(t)} - J^{(t-1)}| \leqslant \epsilon$ or $t>T$
    return $V^{(t)}$, $U^{(t-1)}$
end procedure
```

IV. Experimental Results

In this section, we assess the performance of the proposed method on synthetic and real-world datasets. The measurements used for performance evaluation and the experimental setup are first described. Then, the performance of the proposed algorithms IKG-FCM and TKG-FCM on two 2-D synthetic datasets is reported and discussed. Further comprehensive comparison with seven related algorithms is conducted on 11 public datasets from the UCI machine learning repository [41]. All of the algorithms were implemented in MATLAB, and experiments were run on a computer with a 2.90-GHz CPU and 40-GB RAM.

A. Evaluation Indices and Experimental Setup

To quantify the performance of the proposed algorithm and selected comparative algorithms, we use two major categories of evaluation indices, i.e., hard clustering indices and fuzzy clustering indices. Fuzzy clustering algorithms divide each sample into its cluster corresponding to the maximum membership according to the membership matrix, and thus, hard clustering indices can also be applied to fuzzy clustering algorithms. However, the hard clustering indices ignore the memberships in fuzzy clustering algorithms in such a way that they often fail to faithfully reflect the performance of the fuzzy clustering algorithms [42]. Therefore, we adopt five hard clustering indices and two fuzzy clustering indices to compare the quality of the KG-FCM and seven state-of-the-art algorithms: K-Means, FCM, DPC [31], VFCM, IV-FCM [6], DVPFCM [7] and the evolving fuzzy clustering approach (EFCA) [43].

Four hard clustering indices for assessing crisp partitions are: 1) clustering accuracy (CA) [44], 2) rand index (RI) [45], 3) adjusted rand index (ARI) [46] and 4) normalized mutual information (NMI) [47]. CA, RI and NMI take a value within the interval $[0, 1]$. The higher the values are, the better the clustering performance. The ARI can be negative and has a wider range of values than the RI.

Since most of the adopted methods for comparison are fuzzy clustering algorithms, the fuzzy validity metric is naturally very appropriate. In this research, the extension index of ARI (EARI) [42] and the Xie-Beni (XB) [48] index are used to further evaluate and compare the performance of different fuzzy clustering algorithms. The EARI index rewrites ARI by accounting for the membership values using the basic concepts from set theory. The higher the value of EARI is, the more similar the data structure of the two clusters, and the better the clustering performance. The XB index is a classical and popular metric for measuring the fuzzy clustering performance. Usually, the smaller the XB values are, the better the fuzzy division of the clustering method. The six evaluation indices are described in detail in Appendix-II. EFCA uses K-Means to obtain the final clustering result, and thus there is no XB or EARI for EFCA.

Because the performance of these fuzzy clustering algorithms depends on the initial values, 50 runs of each algorithm with different initializations are implemented under specific parameter settings. Table I shows the parameters of the adopted algorithms. These parameters are commonly used

TABLE I
PARAMETERS SETTING OF THE ADOPTED ALGORITHMS

Algorithms	Parameters
DPC	Radius r: $D(\lceil n * 0.02 \rceil)$ where D is data distance
FCM	Fuzzy index m: 2
V-FCM	Fuzzy index m: same to FCM
IV-FCM	Fuzzy index m: same to FCM
DVPFCM	Fuzzy index m: same to FCM; Typicality index p: 2 Parameters a and b: 1, 1
EFCA	Partitioning magnitude p: 0.2 Epoch cut μ: 0
KG-FCM	Fuzzy index m: same to FCM The influence of granular knowledge ζ: 0.7

values or the optimal values given in the original papers. The SR, RI, ARI, EARI and NMI for each iteration-based algorithm shown as experimental results are the maximum values obtained by running the algorithm 50 times repeatedly, while the XBI is the minimum value of the 50 obtained values. The maximum number of iterations, i.e., T, is 200 for all iteration-based clustering algorithms.

B. Synthetic Datasets

The unclear separation between classes and imbalanced samples among different classes are common challenges to clustering algorithms. To visually illustrate the behavior of granular knowledge formed by the proposed KEG method and to evaluate the performance of our KG-FCM algorithm and other comparative clustering algorithms on addressing such challenges, we create two synthetic datasets in two-dimensional space to enable clustering results to be visually observed and verified. These two datasets are denoted as D1 and D2, respectively. Because the clustering results on D1 and D2 of IKG-FCM and TKG-FCM are the same, their performance on synthetic datasets is reported by KG-FCM.

Example 1: As shown in Fig. 4(a), synthetic dataset D1 includes 1150 samples and has 3 classes. Cluster 1, Cluster 2 and Cluster 3 contain 50, 1000 and 100 samples, respectively. Therefore, the number of samples in Cluster 2 is much larger than that in the other two clusters. The KEG method automatically filters out three knowledge points from the D1 dataset based on the $\delta - \rho$ decision graph in Fig. 4(b). After the knowledge granulation step of KEG, the knowledge domains obtained are shown in Fig. 4(c). Fig. 4(d) demonstrates the clustering results of KG-FCM and FCM, where the dots and squares colored red, yellow and blue are the cluster centers obtained by KG-FCM and FCM, respectively. We can observe that three cluster centers positioned by the FCM algorithm are all in Cluster 1 with the negative impact resulting from the averaging phenomenon. In comparison, the proposed KG-FCM algorithm locates the three cluster centers more accurately under the guidance of granular knowledge points, and these centers are distributed in three different clusters. The guidance track of the granular knowledge for the KG-FCM is represented by the dashed line shown in Fig. 4(d), while the iterative update process of the FCM centers is displayed by solid lines. Comparing the trends of these two lines, it can be seen that the high-density granular knowledge has a good guiding effect on the clustering algorithm to detect potential data structures.

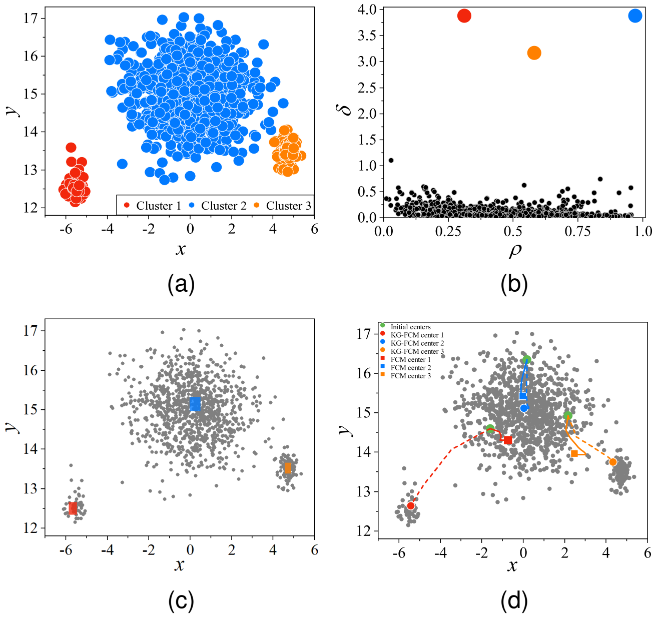


Fig. 4. The synthetic dataset D1. (a) Data distribution. (b) Decision graph of KEG. (c) Granular knowledge location. (d) Clustering results of KG-FCM and FCM.

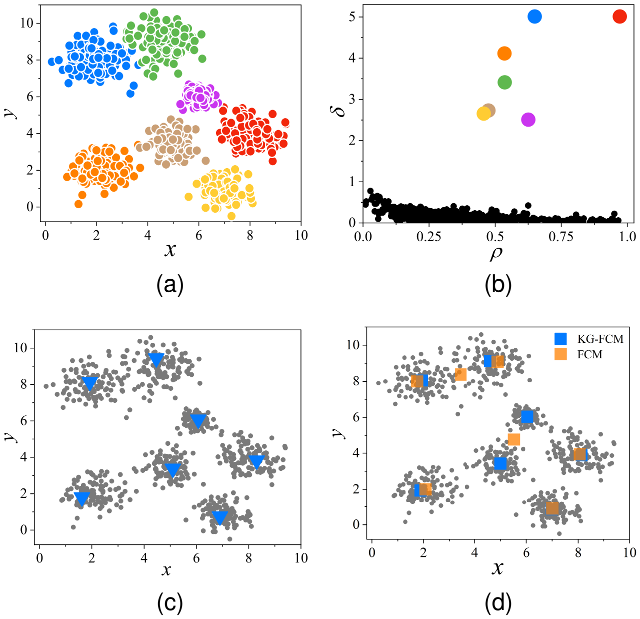


Fig. 5. The synthetic dataset D2. (a) Data distribution. (b) Decision graph of KEG. (C) Knowledge points got by KEG. (d) Clustering results of KG-FCM and FCM.

Example 2: To further verify the accuracy of the proposed KG-FCM algorithm in finding clustering centers, we perform the experiment on the second synthetic dataset D2. The results are shown in Fig. 5. Fig. 5(a) is the distribution of D2. The 7 clusters in D2 have different degrees of overlap and convergence with one another, which brings a large obstacle to the clustering algorithm to successfully locate the cluster centers. From Fig. 5(d), it can be seen that there are two FCM centers in the middle of two clusters owing to the gap between the two adjacent clusters being too close. In addition,

TABLE II
THE CLUSTER CENTERS GOT BY CLUSTERING ALGORITHMS

Algorithms	Centers						
	$v_1^{(Alg)}$	$v_2^{(Alg)}$	$v_3^{(Alg)}$	$v_4^{(Alg)}$	$v_5^{(Alg)}$	$v_6^{(Alg)}$	$v_7^{(Alg)}$
K-Means	(4.1607, 9.4185)	(1.7297, 8.0174)	(2.1669, 2.0372)	(5.1521, 8.7803)	(6.2537, 4.6150)	(3.2662, 7.8024)	(7.2119, 1.2829)
FCM	(8.0659, 3.9321)	(2.0740, 2.0016)	(1.7153, 7.9981)	(4.8179, 9.0733)	(3.2535, 8.2474)	(5.5110, 4.7519)	(7.0113, 0.8900)
V-FCM	(7.8890, 3.8434)	(1.9241, 8.1391)	(2.1840, 2.0509)	(4.8226, 9.2375)	(5.9100, 5.7102)	(4.0101, 8.0724)	(6.9035, 1.0012)
IV-FCM	(8.1278, 3.8838)	(1.9086, 8.1891)	(1.9785, 1.9480)	(4.6319, 9.0817)	(6.0229, 6.0342)	(4.9461, 3.4155)	(7.0190, 0.8621)
DVPFCM	(8.0367, 3.8611)	(1.8837, 8.0000)	(1.8892, 1.9703)	(4.7792, 9.0666)	(5.9804, 5.8182)	(4.9753, 3.4057)	(7.0245, 0.8824)
EFCA	(8.0240, 3.9735)	(2.3103, 8.0344)	(2.1230, 2.0482)	(4.6655, 9.0680)	(5.8769, 6.3649)	(4.8252, 3.3618)	(7.1564, 1.1734)
DPC	(8.7501, 3.7340)	(1.9898, 8.0588)	(2.2379, 1.5605)	(5.6331, 8.7100)	(6.1131, 6.0909)	(4.9379, 2.9913)	(6.9076, 0.7743)
KEG	(8.3089, 3.8247)	(1.9241, 8.1391)	(1.6222, 1.8029)	(4.4664, 9.4400)	(6.0751, 6.0794)	(5.0989, 3.3821)	(6.9003, 0.7528)
KG-FCM	(8.1104, 3.8761)	(1.9762, 8.0260)	(1.9193, 1.9441)	(4.5881, 9.0743)	(6.0343, 6.0324)	(4.9936, 3.4193)	(6.9697, 0.8905)

the cluster centers obtained by the KG-FCM algorithm appear to be located in the densest area of each cluster, which is denser than the dense area where the 7 high-density points obtained by the KEG algorithm are positioned (see Fig. 5(c)).

For the D2 dataset, we further compare the final cluster centers obtained by the KG-FCM and comparative algorithms. The reference cluster centers of D2 are
$V^{(Ref)} = [(8,4);(2,8);(2,2);(4.6,9);(6,6);(5,3.5);(7,0.8)]$.

Table II lists the cluster centers of the clustering algorithms. Then, we can compute the average distance $\bar{d}$ between clustering centers got by algorithms and true centers. The results are shown in Fig. 6. The $\bar{d}$ is calculated as

$$\bar{d} = \frac{\sum_{i=1}^{c} \mathrm{dist}(v_i^{(Alg)}, v_i^{(Ref)})}{c} \tag{34}$$

where $v_i^{(Alg)}$ is the ith center got by the clustering algorithm, and $v_i^{(Ref)}$ is the ith actual center. The results obtained by our proposed KG-FCM algorithm are closest to the true values and its average distance is only 0.0855. There is an interesting result that the average distance metric of V-FCM is more than 8 times that of IV-FCM. This finding means that granular knowledge is more helpful than knowledge points. Furthermore, IV-FCM and DVPFCM perform better than other comparative algorithms on D2. However, these two algorithms both directly use the extracted knowledge as parts of the clustering centers, which leads to overacting, therefore, they are less powerful than our proposed KG-FCM algorithm.

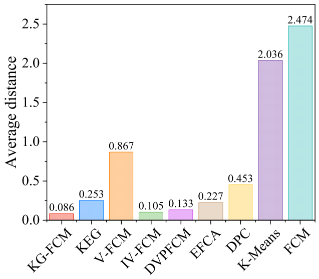


Fig. 6. The average distances between clustering centers and reference centers on the D2 dataset.

C. UCI Datasets

1) Data information:

The UCI database [41] has been widely used by researchers worldwide as a primary source of machine learning datasets since it was created in 1987. In the next experiments, we adopt eleven typical and popularly used UCI datasets to validate the clustering performance of our proposed IKG-FCM and TKG-FCM algorithms, including the Iris, Wine, Glass Identification (Glass), Breast Cancer Wisconsin (BCW), Wholesale Customers (Wholesale), User Knowledge Modeling (UKM), Arcene Training (Arcene), Anuran Calls (MFCC-s), Madelon, Letter (A,B,C,D) Recognition (LR_ABCD) and HTRU2 datasets.

TABLE III
SUMMARY OF THE USED UCI DATASETS

No.	Name	Samples	Features	Classes	size of classes
1	Iris	150	4	3	50,50,50
2	Wine	178	13	3	59,71,48
3	Glass	214	9	6	70,76,17,13,9,29
4	BCW	569	30	2	212,357
5	Wholesale	440	7	2	298,142
6	UKM	258	5	4	24,63,83,88
7	Arcene	100	10000	2	56,44
8	MFCCs	7195	22	10	672,542,3478,310,472, 1121,270,114,68,148
9	Madelon	1999	500	2	999,1000
10	LR_ABCD	3096	16	4	789,766,736,805
11	HTRU2	17898	8	2	16259,1639

The Iris dataset is perhaps the best known database in testing machine learning algorithms. The dataset contains 3 classes of 50 samples each, where one class is linearly separable from the other 2 while the latter are not linearly separable from each other. The Wine, BCW and Arcene datasets have more features than that of Iris, especially the Arcene dataset with 10000 features, which is the highest dimensional dataset. In addition, the common characteristic of the five datasets Glass, Wholesale, UKM, MFCCs and HTRU2 is that the instances among clusters are not balanced. The sample number of the HTRU2 set is 17898, which is the largest in all used datasets. A more detailed introduction of these datasets is summarized in Table III.

2) Experimental results:

In this study, the distance metrics used for the interval and triangular knowledge are different, and as a result, we present two novel clustering algorithms, IKG-FCM and TKG-FCM. The experimental results of these two algorithms are recorded separately in the experiments. For the DPC algorithm, the number of selected cluster centers on the decision graph is as close as possible to the real number on the basis of ensuring that the high density points are a distance away from other points. The setting parameters are shown in Table I.

TABLE IV
THE CA OF DIFFERENT ALGORITHMS FOR DIFFERENT UCI DATASETS

Algorithms	K-Means	DPC	FCM	V-FCM	IV-FCM	DVPFCM	EFCA	IKG-FCM	TKG-FCM
Iris	0.8933 (8)	0.6667 (9)	0.9067 (7)	0.9400 (6)	0.9467 (5)	**0.9600** (1)	**0.9600** (1)	**0.9600** (1)	**0.9600** (1)
Wine	0.9045 (7)	0.8933 (9)	0.9045 (7)	0.9382 (6)	0.9494 (3)	0.9438 (5)	0.9494 (3)	**0.9663** (1)	**0.9663** (1)
Glass	0.4626 (9)	0.4766 (7)	0.4673 (8)	0.5374 (4)	0.5280 (5)	0.5234 (6)	0.5421 (3)	**0.5607** (1)	0.5514 (2)
BCW	0.8295 (9)	0.9104 (5)	0.8401 (8)	0.9209 (3)	0.9174 (4)	0.8840 (7)	0.8998 (6)	**0.9332** (1)	0.9315 (2)
Wholesale	0.7409 (9)	0.8409 (7)	0.7750 (8)	0.8659 (5)	0.8659 (5)	0.8727 (4)	0.8818 (3)	**0.8841** (1)	**0.8841** (1)
UKM	0.4839 (9)	0.4888 (8)	0.4988 (6)	0.4938 (7)	0.5062 (5)	0.5112 (4)	0.5533 (3)	**0.5782** (1)	0.5583 (2)
Arcene	0.5758 (9)	0.6162 (5)	0.6162 (5)	0.6162 (5)	0.6263 (4)	0.6162 (5)	0.6364 (3)	**0.6566** (1)	**0.6566** (1)
MFCCs	0.5576 (9)	0.7404 (4)	0.5600 (8)	0.6126 (7)	0.6461 (5)	0.6428 (6)	0.7607 (3)	**0.8214** (1)	0.8149 (2)
Madelon	0.5023 (6)	0.5458 (4)	0.5183 (5)	0.5003 (9)	0.5008 (7)	0.5008 (7)	0.5798 (3)	**0.6118** (1)	0.5938 (2)
LR_ABCD	0.5223 (9)	0.6505 (3)	0.5310 (8)	0.5985 (6)	0.5869 (7)	0.6295 (5)	0.6318 (4)	**0.6550** (1)	0.6531 (2)
HTRU2	0.9186 (8)	0.7000 (9)	0.9254 (7)	0.9560 (5)	0.9584 (4)	0.9544 (6)	0.9590 (3)	**0.9682** (1)	0.9648 (2)
Arank	8.3636	6.3636	7.0000	5.7273	4.9091	5.0909	3.1818	1.0	1.6364

TABLE V
THE RI OF DIFFERENT ALGORITHMS FOR DIFFERENT UCI DATASETS

Algorithms	K-Means	DPC	FCM	V-FCM	IV-FCM	DVPFCM	EFCA	IKG-FCM	TKG-FCM
Iris	0.8797 (8)	0.7763 (9)	0.8923 (7)	0.9267 (6)	0.9341 (5)	**0.9495** (1)	**0.9495** (1)	**0.9495** (1)	**0.9495** (1)
Wine	0.8781 (7)	0.8654 (9)	0.8775 (8)	0.9175 (6)	0.9324 (3)	0.9261 (5)	0.9311 (4)	**0.9543** (1)	**0.9543** (1)
Glass	0.6944 (8)	0.7030 (7)	0.6931 (9)	0.7310 (3)	0.7167 (5)	0.7053 (6)	0.7187 (4)	**0.7372** (1)	0.7345 (2)
BCW	0.7167 (9)	0.8365 (5)	0.7308 (8)	0.8394 (4)	0.8482 (3)	0.7946 (7)	0.8194 (6)	**0.8751** (1)	0.8721 (2)
Wholesale	0.6430 (9)	0.7318 (7)	0.6505 (8)	0.7672 (5)	0.7672 (5)	0.7773 (4)	0.7911 (3)	**0.7946** (1)	**0.7946** (1)
UKM	0.6592 (8)	0.6275 (9)	0.6889 (6)	0.6933 (5)	0.6732 (7)	0.7047 (4)	0.7050 (3)	0.7103 (2)	**0.7111** (1)
Arcene	0.5065 (9)	0.5222 (5)	0.5222 (5)	0.5222 (5)	0.5271 (4)	0.5222 (5)	0.5325 (3)	**0.5444** (1)	**0.5444** (1)
MFCCs	0.7391 (9)	0.9477 (4)	0.8400 (7)	0.7990 (8)	0.8559 (6)	0.8723 (5)	0.9484 (3)	0.9576 (2)	**0.9588** (1)
Madelon	0.4998 (8)	0.5039 (4)	0.5004 (5)	0.4997 (9)	0.4998 (6)	0.4998 (6)	0.5125 (3)	**0.5248** (1)	0.5174 (2)
LR_ABCD	0.7101 (9)	0.7564 (5)	0.7177 (8)	0.7358 (6)	0.7262 (7)	0.7667 (3)	0.7612 (4)	**0.7744** (1)	0.7709 (2)
HTRU2	0.8504 (8)	0.5799 (9)	0.8618 (7)	0.9158 (6)	0.9203 (4)	0.9129 (5)	0.9230 (3)	**0.9376** (1)	0.9321 (2)
Arank	8.3636	6.6364	7.0909	5.7273	5.0000	4.6364	3.3637	1.1818	1.4545

TABLE VI
THE ARI OF DIFFERENT ALGORITHMS FOR DIFFERENT UCI DATASETS

Algorithms	K-Means	DPC	FCM	V-FCM	IV-FCM	DVPFCM	EFCA	IKG-FCM	TKG-FCM
Iris	0.7302 (8)	0.5681 (9)	0.7560 (7)	0.8341 (6)	0.8512 (5)	0.8857 (2)	**0.8860** (1)	0.8857 (2)	0.8857 (2)
Wine	0.7263 (7)	0.6990 (9)	0.7253 (8)	0.8149 (6)	0.8485 (3)	0.8343 (5)	0.8463 (4)	**0.8975** (1)	**0.8975** (1)
Glass	0.2016 (9)	0.2046 (8)	0.2180 (7)	0.2425 (6)	0.2594 (5)	0.2772 (4)	0.2773 (3)	**0.3381** (1)	0.3362 (2)
BCW	0.4334 (9)	0.6720 (5)	0.4616 (8)	0.6777 (4)	0.6931 (3)	0.5834 (7)	0.6346 (6)	**0.7494** (1)	0.7432 (2)
Wholesale	0.2292 (9)	0.4629 (7)	0.2460 (8)	0.5236 (5)	0.5189 (6)	0.5470 (4)	0.5735 (3)	**0.5804** (1)	**0.5804** (1)
UKM	0.1881 (7)	0.1553 (9)	0.2040 (6)	0.2096 (5)	0.1615 (8)	0.2347 (4)	0.2538 (3)	**0.3055** (1)	0.2838 (2)
Arcene	0.0439 (5)	0.0439 (5)	0.0439 (5)	0.0439 (5)	0.0540 (4)	0.0439 (5)	0.0648 (3)	**0.0888** (1)	**0.0888** (1)
MFCCs	0.4070 (9)	0.8698 (4)	0.5432 (8)	0.5692 (7)	0.5964 (6)	0.7093 (5)	0.8715 (3)	**0.8953** (1)	0.8818 (2)
Madelon	0.0005 (6)	0.0079 (4)	0.0008 (5)	-1E-06 (9)	1E-06 (7)	1E-06 (7)	0.0250 (3)	**0.0495** (1)	0.0347 (2)
LR_ABCD	0.2386 (9)	0.4149 (3)	0.2771 (8)	0.3189 (6)	0.3096 (7)	0.3906 (4)	0.3709 (5)	**0.4178** (1)	0.4153 (2)
HTRU2	0.5320 (8)	0.1074 (9)	0.5800 (7)	0.6313 (5)	0.7289 (2)	0.6148 (6)	0.7191 (4)	**0.7479** (1)	0.7265 (3)
Arank	7.8182	6.5455	7.0000	5.8182	5.0909	4.8182	3.7273	1.0909	1.8182

TABLE VII
THE NMI OF DIFFERENT ALGORITHMS FOR DIFFERENT UCI DATASETS

Algorithms	K-Means	DPC	FCM	V-FCM	IV-FCM	DVPFCM	EFCA	IKG-FCM	TKG-FCM
Iris	0.7419 (8)	0.7337 (9)	0.7550 (7)	0.8192 (6)	0.8449 (5)	0.8642 (4)	**0.8862** (1)	0.8738 (2)	0.8738 (2)
Wine	0.7242 (8)	0.7262 (7)	0.7176 (9)	0.7879 (5)	0.8160 (3)	0.7865 (6)	0.8143 (4)	**0.8662** (1)	**0.8662** (1)
Glass	0.3626 (8)	0.3703 (6)	0.3461 (9)	0.3649 (7)	0.4106 (5)	0.4245 (2)	**0.4265** (1)	0.3970 (3)	0.4143 (4)
BCW	0.3915 (9)	0.5521 (6)	0.3956 (8)	0.5565 (5)	0.6294 (3)	0.5211 (7)	0.5573 (4)	**0.6460** (1)	0.6364 (2)
Wholesale	0.1929 (9)	0.4055 (7)	0.3666 (8)	0.4061 (6)	0.4258 (4)	0.4077 (5)	**0.4703** (1)	0.4411 (3)	0.4482 (2)
UKM	0.2879 (6)	0.1898 (9)	0.2841 (7)	0.3152 (5)	0.2108 (8)	0.3423 (4)	0.3760 (3)	**0.4224** (1)	0.4172 (2)
Arcene	0.0398 (6)	0.0320 (7)	0.0320 (7)	0.0320 (7)	0.0637 (5)	**0.0901** (1)	0.0712 (4)	0.0875 (2)	0.0875 (2)
MFCCs	0.6287 (9)	0.7170 (3)	0.6590 (8)	0.6653 (6)	0.6669 (5)	0.6624 (7)	0.7080 (4)	**0.7683** (1)	0.7497 (2)
Madelon	0.0212 (3)	0.0061 (5)	0.0010 (6)	0.0010 (6)	0.0010 (6)	0.0010 (6)	0.0195 (4)	**0.0364** (1)	0.0256 (2)
LR_ABCD	0.3439 (9)	0.4788 (3)	0.4052 (7)	0.4225 (6)	0.3619 (8)	0.4697 (4)	0.4502 (5)	**0.5323** (1)	0.5261 (2)
HTRU2	0.4698 (7)	0.0820 (9)	0.4332 (8)	0.5047 (5)	0.5444 (4)	0.4903 (6)	0.5494 (3)	**0.5868** (1)	0.5637 (2)
Arank	7.4545	6.4545	7.6364	5.8182	5.0909	4.5454	3.0909	1.5455	2.0909

TABLE VIII
CLUSTER NUMBER OR KNOWLEDGE NUMBER GOT BY DPC AND KEG

Datasets	Iris	Wine	Glass	BCW	Wholesale	UKM	Arcene	MFCCs	Madelon	LR_ABCD	HTRU2
DPC	2	***3***	5	***2***	***2***	***4***	4	11	***2***	***4***	***2***
KEG	2	***3***	4	***2***	***2***	***4***	***2***	***10***	***2***	***4***	***2***

TABLE IX
THE EARI AND XB OF DIFFERENT FUZZY CLUSTERING ALGORITHMS FOR DIFFERENT UCI DATASETS

Algorithms	FCM		V-FCM		IV-FCM		DVPFCM		IKG-FCM		TKG-FCM	
	EARI	XB	EARI	XB	EARI	XB	EARI	XB	EARI	XB	EARI	XB
Iris	0.7740 (6)	0.3204 (6)	0.8750 (5)	0.3075 (5)	0.8810 (4)	0.2794 (4)	0.9404 (3)	**0.2195** (1)	0.9450 (2)	0.2302 (3)	**0.9460** (1)	0.2299 (2)
Wine	0.8280 (6)	0.4784 (6)	0.8875 (5)	0.4397 (5)	0.9032 (4)	**0.2886** (1)	0.9160 (3)	0.3870 (4)	0.9437 (2)	0.3804 (3)	**0.9438** (1)	0.2935 (2)
Glass	0.4653 (6)	0.2944 (5)	0.4764 (5)	0.2841 (3)	0.5564 (3)	0.2962 (6)	0.5305 (4)	0.2858 (4)	**0.6672** (1)	**0.2206** (1)	0.6324 (2)	0.2670 (2)
BCW	0.6450 (6)	0.6816 (6)	0.7200 (5)	0.4569 (4)	0.7681 (3)	0.4085 (3)	0.7385 (4)	0.4710 (5)	0.8787 (2)	**0.2068** (1)	**0.8790** (1)	0.2945 (2)
Wholesale	0.4020 (6)	0.7577 (6)	0.5585 (4)	0.5885 (5)	0.5965 (3)	0.4175 (4)	0.5510 (5)	0.4001 (3)	**0.7951** (1)	**0.2025** (1)	**0.7951** (1)	0.2751 (2)
UKM	0.3968 (6)	0.7017 (4)	0.4176 (5)	**0.6543** (1)	0.4297 (4)	0.7906 (6)	0.4595 (3)	0.7747 (5)	**0.3621** (1)	0.6832 (3)	0.3368 (2)	0.6760 (2)
Arcene	0.1330 (6)	0.7866 (6)	0.3588 (4)	0.7247 (5)	0.3685 (3)	0.6146 (4)	0.3468 (5)	0.5845 (3)	0.5047 (2)	**0.4625** (1)	**0.5071** (1)	0.4967 (2)
MFCCs	0.5030 (5)	1.1704 (6)	0.4935 (6)	**0.2190** (1)	0.6774 (4)	0.4155 (5)	0.8494 (3)	0.3006 (4)	**0.9410** (1)	0.2532 (3)	0.9323 (2)	0.2508 (2)
Madelon	0.0334 (4)	18.775 (6)	0.0015 (5)	0.7861 (4)	0.1683 (3)	0.8715 (5)	0.0012 (6)	0.6719 (3)	**0.4361** (1)	**0.0644** (1)	0.3387 (2)	0.0657 (2)
LR_ABCD	0.4069 (6)	1.3183 (6)	0.4641 (5)	0.4947 (3)	0.5221 (4)	0.5696 (4)	0.5230 (3)	1.0081 (5)	**0.7070** (1)	**0.2140** (1)	0.6451 (2)	0.2477 (2)
HTRU2	0.4500 (6)	0.2869 (6)	0.4790 (5)	0.2843 (5)	0.5006 (3)	0.1340 (3)	0.4980 (4)	0.1107 (2)	0.7321 (2)	**0.0980** (1)	**0.7323** (1)	0.1615 (4)
Arank	5.7	5.7	4.9	3.7	3.5	4.1	3.9	3.5	1.5	1.7	1.5	2.2

TABLE X
AVERAGE RUN TIME OF 8 ALGORITHMS ON 11 UCI DATASETS

Datasets	K-Means	FCM	V-FCM	IV-FCM	DVPFCM	EFCA	IKG-FCM	TKG-FCM
Iris	**0.0033** (1)	0.0535 (7)	0.0294 (5)	0.0313 (6)	0.0151 (4)	0.0886 (8)	0.0065 (2)	0.0110 (3)
Wine	**0.0030** (1)	0.0242 (7)	0.0188 (5)	0.0203 (6)	0.0168 (4)	0.1134 (8)	0.0134 (2)	0.0158 (3)
Glass	**0.0036** (1)	0.0706 (7)	0.0530 (5)	0.0629 (6)	0.0439 (4)	0.3306 (8)	0.0430 (3)	0.0379 (2)
BCW	**0.0066** (1)	0.0851 (7)	0.0446 (5)	0.0468 (6)	0.0360 (4)	0.3168 (8)	0.0341 (3)	0.0302 (2)
Wholesale	**0.0049** (1)	0.0674 (7)	0.0337 (6)	0.0255 (5)	0.0252 (4)	0.3047 (8)	0.0132 (2)	0.0169 (3)
UKM	**0.0064** (1)	0.1684 (6)	0.2120 (7)	0.1649 (5)	0.1362 (4)	0.4020 (8)	0.1125 (3)	0.0823 (2)
Arcene	**0.0301** (1)	1.1063 (7)	0.6585 (6)	0.6159 (4)	0.6539 (5)	3.4573 (8)	0.5507 (3)	0.3908 (2)
MFCCs	**0.1635** (1)	2.4885 (4)	0.7022 (2)	2.8061 (6)	2.3488 (5)	6.9323 (8)	1.0184 (3)	2.8780 (7)
Madelon	0.0581 (2)	0.2559 (6)	**0.0264** (1)	0.3241 (7)	0.1978 (4)	1.2919 (8)	0.1835 (3)	0.2174 (5)
LR_ABCD	**0.0276** (1)	0.8063 (7)	0.1469 (2)	0.4173 (6)	0.3370 (5)	2.5401 (8)	0.2900 (4)	0.2650 (3)
HTRU2	0.2489 (3)	1.5348 (6)	0.0967 (2)	0.4748 (5)	2.9624 (7)	7.1977 (8)	**0.0927** (1)	0.2651 (4)
Arank	1.27	6.45	4.18	5.64	4.55	8	2.64	3.27

The CA, RI, ARI and NMI of each algorithm on different UCI datasets are shown in Tables IV, V, VI and VII. The EARI and XB values of the fuzzy clustering algorithms are recorded in Table IX. These indices values are the best results obtained from 50 repeated runs on algorithms (except for the DPC algorithm). The best results are in bolded in Table IV to IX. Next to each validity is the relative ranking of the algorithm on the dataset. The Arank of each table is the average rank of the algorithms on all UCI datasets used.

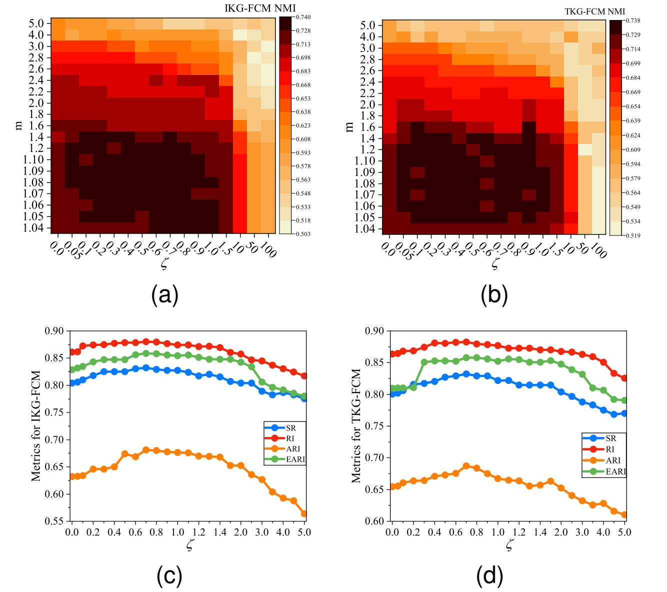
As shown in Table IV to Table VII, the comparison experiments on 11 UCI datasets demonstrate that our proposed IKG-FCM and TKG-FCM predominantly outperform the other seven clustering algorithms in terms of CA, RI, ARI and NMI, except for the Arcene dataset. Among 44 evaluation parameters (11 datasets-by-4 measure indices), IKG-FCM championed 37 times consistently on all 11 datasets, while it ranked second five times and third two times. For the TKG-FCM algorithm, its performance is slightly inferior to that of IKG-FCM. However, in 44 comparisons, TKG-FCM performs better than the other algorithms 42 times (excluding IKG-FCM). This observation shows that interval knowledge is more conducive to clustering than triangular knowledge. Interval knowledge represents the entire density area without emphasizing the representative points as TKG-FCM does. This approach weakens the negative effects of the strong guidance of the knowledge points.

In addition, for the significantly unbalanced datasets Glass and Anuran Calls (MFCCs), the clustering validity of the KG-FCM (IKG-FCM and TKG-FCM) algorithms is noticeably superior to other clustering algorithms. Table IV indicates more than a 17% improvement on Glass (from 0.4626 to 0.5607) and 47% for MFCCs (from 0.5576 to 0.8214) in comparison to the K-Means method. In addition, when accounting for the average rank of each algorithm, it can be seen that the performance relationship of the clustering algorithms is as follows:
K-Means < FCM < DPC < V-FCM < IV-FCM < DVPFCM < EFCA < TKG-FCM < IKG-FCM.

The performance of IV-FCM is better than that of V-FCM and sometimes even outperforms DVPFCM, which proves that the interval viewpoint possesses more positive guidance. The DVPFCM algorithm considers the typicality of data to clusters,and thus, it is more robust. However, the V-FCM, IV-FCM and DVPFCM all output knowledge points directly as parts of clustering centers, which severely limits the flexibility of the algorithms. On average, the DPC algorithm performs slightly worse. This result could occur because the number of clusters is not determined correctly (see Table VIII).

The knowledge number of KEG is shown in Table VIII. We keep the number of knowledge points equal to the reference number of clusters in bold. Compared to the actual cluster number of each set in Table III, KEG sometimes extracts excess or less high-density knowledge than the true number of clusters. However, the proposed IKG-FCM and TKG-FCM algorithms control the influence of granular knowledge through parameter ω to maximize the positive guidance of knowledge and weaken its incorrect supervision.

As seen from Table IX, IKG-FCM and TKG-FCM outperform the other four fuzzy clustering algorithms (not including EFCA) in terms of both EARI and XB. This finding occurs mainly because the two algorithms, IKG-FCM and TKG-FCM, not only predetermine the influence of the samples on the clustering according to the high-density points extracted by KEG but also locate the cluster centers more accurately with the help of granular knowledge. Thus the fuzziness of



(a)

(b)

(c)

(d)

Fig. 7. Analysis of the parameters setting on results on Iris, Wine and Glass dataset for IKG-FCM and TKG-FCM. (a) Average best NMI indices variation with regard to m and ζ for IKG-FCM, (b) Average best NMI indices variation with regard to m and ζ for TKG-FCM, (c) Effect of the parameter ζ on IKG-FCM, (d) Effect of the parameter ζ on TKG-FCM.

the cluster partitions by IKG-FCM and TKG-FCM are lower than those of the other four clustering algorithms.

3) *Parameter Analyses:*

In IKG-FCM and TKG-FCM, there are two parameters, including the fuzziness degree m and the parameter ζ to balance the influence of granular knowledge, which must be set in advance. To analyze the effect of parameters m and ζ on the clustering performance of IKG-FCM and TKG-FCM, we present the results on the Iris, Wine and Glass datasets.

Fig. 7(a) and (b) display average best NMI indices for various values of m and ζ, where each color corresponds to a different average best NMI. And the darker the red, the higher the average best NMI value. So, from Fig. 7(a) and (b), we deduce that the performance of IKG-FCM and TKG-FCM reach the best when m is in $[1.05, 1.2]$, and ζ is in $[0.2, 1.0]$. Overall, the IKG-FCM algorithm performs better than TKG-FCM because the darker red coverage area of Fig. 7(a) is broader than Fig. 7(b). This result has also been observed for Tables IV to IX. Moreover, when m is in $[1.04, 2.4]$, the performances of IKG-FCM and TKG-FCM are not very sensitive to the value of m, while parameter ζ does not exceed 1.0. Their obtained average best NMI values are all above 0.68 and differ from the maximum value of 0.74 by no more than 0.06.

Choosing the best m remains an open problem for fuzzy clustering [49]–[51]. A value of 2.0 for m is commonly used in fuzzy clustering algorithms, and thus, we fix the value of m at 2.0 in this study [50], [52]. If a slightly better algorithm performance is sought, $m = 1.08$ could be a choice. Experimental results in [51] show that the fuzzy clustering algorithms achieve the best performance when m is taken around 1.08.

We further compare the impacts of ζ on the clustering performance with different values. Fig. 7(c) and (d) illustrate the changes in the four evaluation metrics with regard to $\zeta \in [0, 5.0]$ when m is fixed as 2.0. Fig. 7(c) and (d) show that when ζ is set to 0.7, our IKG-FCM and TKG-FCM reach their peak performances.

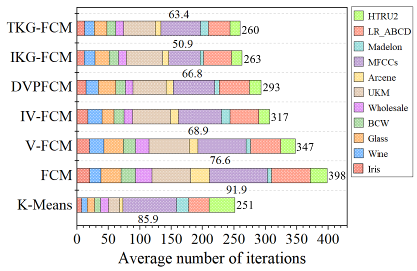


Fig. 8. The average iterations of 7 algorithms on 11 UCI datasets.

4) *Iterations and run time of each algorithm:*

With the same initialization, the iteration runs and run time of each algorithm are compared on the 11 tested UCI datasets. EFCA runs on multiple epochs of a dataset, and in each epoch, only a portion of the data are clustered by K-Means. Therefore, we do not compare the iterations of EFCA with other algorithms. As shown in Fig. 8, overall, except for K-Means, TKG-FCM and IKG-FCM use the fewest runs (259.6 and 262.5, respectively) in comparison to the other four algorithms to achieve convergence on the datasets. In particular, TKG-FCM and IKG-FCM have the fewest iterations on the MFCCs dataset and even outperform K-Means.

Table X presents the running time of each algorithm on the used UCI datasets. Combining the iterations of each algorithm shown in Fig. 8, IKG-FCM runs faster than TKG-FCM when their iteration numbers are similar. This finding occurs because the distance calculation between triangular data is more complicated. EFCA is the most time-consuming because it requires multiple runs of FCM to complete clustering. The IV-FCM algorithm runs slower than V-FCM because the distance calculation between interval data is more time-consuming than the calculation of the Euclidean distance. In addition, the DVPFCM involves the calculation of the typicality matrix, which increases its running time and makes it slower than the V-FCM algorithm. Overall, the running time of each algorithm is ordered as: K-Means<IKG-FCM<TKG-FCM<V-FCM<IV-FCM<FCM<DVPFCM<EFCA.

5) *Friedman and post-hoc tests:*

To check whether significant differences exist between the proposed IKG-FCM, TKG-FCM algorithms and comparative methods, Frideman [53] and Nemenyi post-hoc test [54] are utilized. The significance levels of both tests are set to 5%. The last row in Tables IV-VII presents the average rank of the different algorithms. With $k = 9$ algorithms and $N = 11$ datasets, the results of the Friedman test are computed as follows:

$$\begin{aligned}\tau_{\chi^2} &= \frac{12N}{k(k+1)}\Big(\sum_{i=1}^{k} r_i^2 - \frac{k(k+1)^2}{4}\Big),\\ \tau_F &= \frac{(N-1)\tau_{\chi^2}}{N(k-1)-\tau_{\chi^2}}\end{aligned} \tag{35}$$

where r_i is the average rank of the ith algorithm. The results are presented in Table XI. τ_F is subject to the F distribution with the degree of freedom $k-1=8$ and $(k-1)(N-1)=80$. For the significance level $\alpha = 0.05$, the critical value of

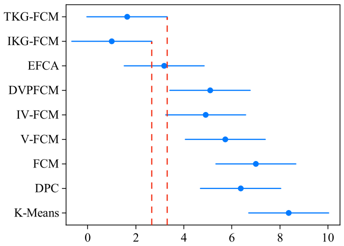


Fig. 9. Nemenyi test based on CA in Table IV.

$F(8,80)$ is $2.056 < \tau_F$, which indicates that the methods are statistically different.

TABLE XI
RESULTS OF FRIEDMAN TEST

Friedman Test	CA	RI	ARI	NMI
τ_F	45.5625	48.9828	43.5293	66.8231

We further discriminate the difference of these algorithms with a Nemenyi post-hoc test by sorting the CA in the above experiments, which states that the performance of the two methods is significantly different if the ranking difference between two algorithms is larger than the critical threshold CD. CD is computed by (36).

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}. \tag{36}$$

We can obtain $q_\alpha = 2.855$ from the Nemenyi table when the confidence level $\alpha = 0.1$. Then, $CD = 3.3339$ is obtained. Fig. 9 shows the results of the Nemenyi test, where the x axis shows the value of the average rankings and the y axis corresponds to 9 algorithms. For each algorithm, its average ranking is represented by a blue point and its confidence interval is shown by a blue line with length CD. The two red dotted lines show the upper confidence levels of IKG-FCM and TKG-FCM. If the vertical lines of two algorithms have an overlap, the two algorithms are not significantly different. From Fig. 9, it is clearly shown that IKG-FCM and TKG-FCM obtain a lower average rank than the other 7 algorithms. IKG-FCM and TKG-FCM are significantly different from K-Means, DPC, FCM, V-FCM, IV-FCM and DVPFCM. They have a relatively small difference with EFCA.

V. CONCLUSIONS

In this study, we first propose a new granular knowledge extraction and granulation method KEG. We then introduce the knowledge-based clustering mechanism to develop two versions of KG-FCM methods based on the types of granular knowledge obtained by KEG. KEG has the characteristic of automatically identifying high-density points by the three-sigma rule. In addition, the selected high-density points are extended to granular knowledge by merging these points' natural neighbors. We adopted two forms to express granular knowledge. One is the interval data, while the other is triangular fuzzy numbers. Therefore, two KG-FCM methods are proposed: IKG-FCM and TKG-FCM. In both objective functions, a knowledge role term is added to guide the clustering procedure to find the potential data structure. The superiorities of the KG-FCM algorithms can be summarized as follows:

1) Compared to existing knowledge-based clustering algorithms (V-FCM, IV-FCM and DVPFCM), parts of prototypes are not directly replaced by domain knowledge. In IKG-FCM and TKG-FCM, the clustering centers are updated with the guidance of granular knowledge during iterations, which avoids the excessive influence of knowledge and improves the flexibility of the algorithms.
2) Domain knowledge is more representative. The interval or triangular knowledge provided by the proposed KEG method is located in high-density areas because their corresponding origin points have a higher density compared to other points. Therefore, the KG-FCM algorithms can grasp the high-density property of granular knowledge to find clustering centers more correctly.
3) The effectiveness of IKG-FCM and TKG-FCM was also proven by the experiments on synthetic and real datasets. The experimental results show that our methods achieve better performance in the distance between clustering centers and reference centers and various evaluation indices of some of the datasets, especially imbalanced datasets. Moreover, comparing the clustering time and iterations of our methods with other comparative algorithms, we find that the guidance of granular knowledge speeds up the convergence of the algorithms. In general, the results of IKG-FCM are slightly better than those of TKG-FCM. This finding demonstrates that not emphasizing a point of an area is of help to the clustering algorithm.

In future studies, it will be worthwhile to reduce the computing complexity of the KEG knowledge extraction method to make our proposed knowledge-based clustering algorithms more efficient on large-scale data. Furthermore, most clustering algorithms need to assign the number of clusters before processing. IKG-FCM and TKG-FCM are not exceptions. Thus, further discovering KG-FCM algorithms by automatically finding the optimal number of clusters could be a future investigation.

REFERENCES

[1] J. C. Dunn, "A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.

[2] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. Plenum Press, 1981.

[3] H. Li and M. Wei, "Fuzzy clustering based on feature weights for multivariate time series," *Knowledge-Based Systems*, vol. 197, p. 105907, 2020.

[4] K. Honda, K. Hayashi, S. Ubukata, and A. Notsu, "Fuzzy-possibilistic clustering for categorical multivariate data," in *the 60th Annual Conference of the Society of Instrument and Control Engineers of Japan*. IEEE, 2021, pp. 9–14.

[5] C. Wu and X. Zhang, "A novel kernelized total bregman divergence-driven possibilistic fuzzy clustering with multiple information constraints for image segmentation," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 6, pp. 1624–1639, June 2022.

[6] W. Pedrycz, V. Loia, and S. Senatore, "Fuzzy clustering with viewpoints," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 2, pp. 274–284, April 2010.

[7] Y. Tang, X. Hu, W. Pedrycz, and X. Song, "Possibilistic fuzzy clustering with high-density viewpoint," *Neurocomputing*, vol. 329, pp. 407–423, 2019.

[8] W. Dai, Q. Yang, G. Xue, and Y. Yu, "Self-taught clustering," in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 200–207.

[9] W. Jiang and F. Chung, "Transfer spectral clustering," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2012, pp. 789–803.

[10] Z. Deng, Y. Jiang, F. Chung, H. Ishibuchi, K. Choi, and S. Wang, "Transfer prototype-based fuzzy clustering," *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 5, pp. 1210–1232, October 2015.

[11] S. Eden, "Environmental issues: Knowledge, uncertainty and the environment," *Progress in Human Geography*, vol. 22, no. 3, pp. 425–432, 1998.

[12] W. Pedrycz, *Knowledge-based clustering: From data to information granules*. John Wiley & Sons, 2005.

[13] R. J. Hathaway, J. C. Bezdek, and W. Pedrycz, "A parametric model for fusing heterogeneous fuzzy data," *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 3, pp. 270–281, August 1996.

[14] S. FrühwirthSchnatter, "On statistical inference for fuzzy data with applications to descriptive statistics," *Fuzzy Sets and Systems*, vol. 50, no. 2, pp. 143–165, 1992.

[15] M. Yang and C. Ko, "On a class of fuzzy c-numbers clustering procedures for fuzzy data," *Fuzzy Sets and Systems*, vol. 84, no. 1, pp. 49–60, 1996.

[16] M. Yang and H. Liu, "Fuzzy clustering procedures for conical fuzzy vector data," *Fuzzy Sets and Systems*, vol. 106, no. 2, pp. 189–200, 1999.

[17] W. Hung and M. Yang, "Fuzzy clustering on LR-type fuzzy numbers with an application in Taiwanese tea evaluation," *Fuzzy Sets and Systems*, vol. 150, no. 3, pp. 561–577, 2005.

[18] W. Hung, M. Yang, and E. S. Lee, "A robust clustering procedure for fuzzy data," *Computers & Mathematics with Applications*, vol. 60, no. 1, pp. 151–165, 2010.

[19] M. B. Ferraro and P. Giordani, "On possibilistic clustering with repulsion constraints for imprecise data," *Information Sciences*, vol. 245, pp. 63–75, 2013.

[20] P. DUrso and P. Giordani, "A weighted fuzzy c-means clustering model for fuzzy data," *Computational Statistics & Data Analysis*, vol. 50, no. 6, pp. 1496–1523, 2006.

[21] R. Coppi, P. DUrso, and P. Giordani, "Fuzzy and possibilistic clustering for fuzzy data," *Computational Statistics & Data Analysis*, vol. 56, no. 4, pp. 915–927, 2012.

[22] P. D'Urso and L. De Giovanni, "Robust clustering of imprecise data," *Chemometrics and Intelligent Laboratory Systems*, vol. 136, pp. 58–80, 2014.

[23] A. Gacek and W. Pedrycz, "Clustering granular data and their characterization with information granules of higher type," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 4, pp. 850–860, August 2014.

[24] S. Auephanwiriyakul and J. M. Keller, "Analysis and efficient implementation of a linguistic fuzzy c-means," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 5, pp. 563–582, October 2002.

[25] P. Grzegorzewski, "Metrics and orders in space of fuzzy numbers," *Fuzzy Sets and Systems*, vol. 97, no. 1, pp. 83–94, 1998.

[26] A. Gacek and W. Pedrycz, "Clustering granular data and their characterization with information granules of higher type," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 4, pp. 850–860, August 2015.

[27] Y. Shen, W. Pedrycz, and X. Wang, "Clustering homogeneous granular data: Formation and evaluation," *IEEE Transactions on Cybernetics*, vol. 49, no. 4, pp. 1391–1402, April 2019.

[28] Q. Zhu, J. Feng, and J. Huang, "Natural neighbor: A self-adaptive neighborhood method without parameter k," *Pattern Recognition Letters*, vol. 80, pp. 30–36, 2016.

[29] R. A. Jarvis and E. A. Patrick, "Clustering using a similarity measure based on shared near neighbors," *IEEE Transactions on Computers*, vol. 100, no. 11, pp. 1025–1034, November 1973.

[30] K. Chidananda Gowda and G. Krishna, "Agglomerative clustering using the concept of mutual nearest neighbourhood," *Pattern Recognition*, vol. 10, no. 2, pp. 105–112, 1978.

[31] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[32] M. Du, S. Ding, and H. Jia, "Study on density peaks clustering based on k-nearest neighbors and principal component analysis," *Knowledge-Based Systems*, vol. 99, pp. 135–145, 2016.

[33] S. A. Seyedi, A. Lotfi, P. Moradi, and N. N. Qader, "Dynamic graph-based label propagation for density peaks clustering," *Expert Systems with Applications*, vol. 115, pp. 314–328, 2019.

[34] X. Xu, S. Ding, and Z. Shi, "An improved density peaks clustering algorithm with fast finding cluster centers," *Knowledge-Based Systems*, vol. 158, pp. 65–74, 2018.

[35] R. Mehmood, G. Zhang, R. Bie, H. Dawood, and H. Ahmad, "Clustering by fast search and find of density peaks via heat diffusion," *Neurocomputing*, vol. 208, pp. 210–217, 2016.

[36] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," in *the 21st International Conference on Data Engineering*, vol. 1, no. 1, 2005, pp. 341–352.

[37] F. Pukelsheim, "The three sigma rule," *The American Statistician*, vol. 48, no. 2, pp. 88–91, 1994.

[38] A. Kvanli, R. Pavur, and K. Keeling, *Concise managerial statistics*. Cengage Learning, 2005.

[39] M. R. Chernick, *The essentials of biostatistics for physicians, nurses, and clinicians*. John Wiley & Sons, 2011.

[40] R. Lan and J. Fan, "A fuzzy c-means type clustering algorithm on triangular fuzzy numbers," in *the 6th International Conference on Fuzzy Systems and Knowledge Discovery*. IEEE, 2009, pp. 12–16.

[41] D. Dua and C. Graff, "UCI machine learning repository," 2017, http://archive.ics.uci.edu/ml.

[42] R. J. Campello, "A fuzzy extension of the rand index and other related indexes for clustering and classification assessment," *Pattern Recognition Letters*, vol. 28, no. 7, pp. 833–841, 2007.

[43] A. S. Shirkhorshidi, T. Y. Wah, S. M. R. Shirkhorshidi, and S. Aghabozorgi, "Evolving fuzzy clustering approach (EFCA): An epoch clustering that enables heuristic post pruning," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 3, pp. 560–568, March 2021.

[44] X. Zhu, W. Pedrycz, and Z. Li, "Fuzzy clustering with nonlinearly transformed data," *Applied Soft Computing*, vol. 61, pp. 364–376, 2017.

[45] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, 1971.

[46] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193–218, 1985.

[47] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research*, vol. 3, no. 3, pp. 583–617, 2002.

[48] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 841–847, 1991.

[49] D. Graves and W. Pedrycz, "Kernel-based fuzzy clustering and fuzzy clustering: A comparative experimental study," *Fuzzy Sets and Systems*, vol. 161, no. 4, pp. 522–543, 2010.

[50] K. L. Wu, "Analysis of parameter selections for fuzzy c-means," *Pattern Recognition*, vol. 45, no. 1, pp. 407–415, 2012.

[51] H. C. Huang, Y. Y. Chuang, and C. S. Chen, "Multiple kernel fuzzy clustering," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 1, pp. 120–134, February 2012.

[52] R. L. Cannon, J. V. Dave, and J. C. Bezdek, "Efficient implementation of the fuzzy c-means clustering algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 2, pp. 248–255, March 1986.

[53] D. W. Zimmerman and B. D. Zumbo, "Relative power of the Wilcoxon test, the Friedman test, and repeated-measures ANOVA on ranks," *The Journal of Experimental Education*, vol. 62, no. 1, pp. 75–86, 1993.

[54] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[55] F. Werner and Y. N. Sotskov, *Mathematics of Economics and Business*. Taylor & Francis, 2006.

[56] N. Hindman, "Basically bounded sets and a generalized Heine-Borel theorem," *The American Mathematical Monthly*, vol. 80, no. 5, pp. 549–552, 1973.

Xianghui Hu received the B.Eng. Degree in internet of things in 2016 and the M.Eng. Degree in computer application technology in 2019, both from the Hefei University of Technology, China. She is now a Ph.D Candidate in the area of social computing with Southeast University, China. Her research direction lies in social computing and pattern recognition with application to image processing. She serves as a reviewer of *PLOS ONE, Information Sciences* and *IEEE Access*.

Yiming Tang (Member, IEEE) received the Ph.D. degree from Hefei University of Technology, Hefei, China, in 2011. He is currently an Associate Professor with Hefei University of Technology and is also a Visiting Professor with University of Alberta, Edmonton, AB, Canada. He has authored or coauthored more than 70 academic papers. His research interests include clustering, fuzzy inference, fuzzy system, granular computing, image processing, and affective computing. He is an Associate Editor for *Information Sciences*, *IEEE ACCESS* and *Journal of Intelligent & Fuzzy Systems*, a Member of Editorial Board of *PLOS ONE*, and a Senior Member of China Computer Federation (CCF), and also a Senior Member of the Chinese Association for Artificial Intelligence (CAAI). He is a Professional Committee of Machine Learning of CAAI, a Professional Committee of Granular Computing and Knowledge Discovery of CAAI, a Professional Committee of Cooperative Computing of CCF, and a Professional Committee of Non-classical Logic and Computation of Chinese Association of Logic. He is a reviewer of more than 30 journals and conferences.

Witold Pedrycz (Life Fellow, IEEE) received the M. Sc., Ph. D., and D. Sc. degrees from Silesian University of Technology, Gliwice, Poland, in 1977, 1980, and 1984, respectively. He is a Professor and the Canada Research Chair of computational intelligence with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. His current research interests include computational intelligence, fuzzy modeling and granular computing, knowledge discovery and data mining, fuzzy control, software engineering. He is also the author of 14 research monographs that cover various aspects of computational intelligence and software engineering. He is an Editor-in-Chief of *Information Sciences*. He is also an Associate Editor of *IEEE Transactions on Fuzzy Systems*, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. He has been a member of numerous program committees of IEEE conferences in the area of fuzzy sets and neurocomputing. He received the prestigious Norbert Wiener Award from the IEEE Systems, Man, and Cybernetics Council in 2007. He received the IEEE Canada Computer Engineering Medal in 2008 and the Cajastur Prize for Soft Computing in 2009 from the European Centre for Soft Computing for pioneering and multifaceted contributions to granular computing. He is a Fellow of the Royal Society of Canada.

Kai Di received the B.S. degree at the School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China, in 2016. He is currently pursuing the Ph.D degree at the Intelligent Systems and Social Computing Laboratory, School of Computer Science and Engineering, Southeast University, Nanjing, China. His current research interest is multi-agent systems.

Jiuchuan Jiang received the Ph.D degree in Computer Science from Nanyang Technological University, Singapore in 2019. He is now with the School of Information Engineering, Nanjing University of Finance and Economics, Nanjing, China. His main research interests include clustering, social networks, crowdsourcing, and multiagent systems. He has published several scientific articles in refereed journals and conference proceedings, such as the *IEEE Transactions on Parallel and Distributed Systems*, the *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, the *ACM Transactions on Autonomous and Adaptive Systems*, and the International Conference on Autonomous Agents and Multiagent Systems (AAMAS).

Yichuan Jiang (Senior Member, IEEE) received the Ph.D. degree in computer science from Fudan University, Shanghai, China, in 2005. He is currently a Distinguished Professor and the Director of the Laboratory of Intelligent Systems and Social Computing, School of Computer Science and Engineering, Southeast University, Nanjing, China. He has published more than 100 scientific articles in refereed journals such as the *IEEE Transactions on Parallel and Distributed Systems*, the *IEEE Journal on Selected Areas in Communications*, the *IEEE Transactions on Mobile Computing*, the *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, the *IEEE Transactions on Cybernetics*, the *ACM Transactions on Autonomous and Adaptive Systems*, the *ACM Transaction on Intelligent Systems and Technology*, the *Journal of Autonomous Agents and Multi-Agent Systems*, and in conference proceedings such as IJCAI, AAAI and AAMAS. His main research interests include multiagent systems, social computing, and social networks.